Subsampled Factor Models for Asset Pricing: The Rise of Vasa

Gianluca De Nard
Department of Finance
New York University
New York, NY 10012, USA
denard@stern.nyu.edu

Simon Hediger
Department of Banking and Finance
University of Zurich
CH-8032 Zurich, Switzerland
simon.hediger@bf.uzh.ch

Markus Leippold
Department of Banking and Finance
University of Zurich
CH-8032 Zurich, Switzerland
markus.leippold@bf.uzh.ch

March 2020

Abstract

We propose a new method, VASA, based on variable subsample aggregation of model predictions for equity returns using a large-dimensional set of factors. To demonstrate the effectiveness, robustness, and dimension reduction power of VASA, we perform a comparative analysis between state-of-the-art machine learning algorithms. As a performance measure, we explore not only the global predictive but also the stock-specific R^2's and their distribution. While the global R^2 indicates the average forecasting accuracy, we find that high variability in the stock-specific R^2's can be detrimental for the portfolio performance, due to the higher prediction risk. Since VASA shows minimal variability, portfolios formed on this method outperform the portfolios based on more complicated methods like random forests and neural nets.

KEY WORDS: Large-dimensional factor models, machine learning, return prediction, subagging, subsampling.

JEL CLASSIFICATION NOS: C13, C30, C53, C58, G12, G17.

1 Introduction

Machine learning has had a triumphal march over the last decade, showing unprecedented success in a great variety of different scientific fields. As the quest for factors to explain the cross-section of equity returns has produced a zoo of factors,[1] machine learning seems to be ideally suited to tame the curse of dimensionality when factor models are employed to predict equity returns. Hence, it appears that machine learning methods set out to conquer the world of finance as well. Indeed, recent research suggests that machine learning models dominate traditional models in predicting cross-sectional stock returns. In a landmark paper, Gu et al. (2020) provide a comprehensive comparison of different machine learning methods for predicting the cross-section of individual US stock returns and find that neural networks and random forests emerge as the methods of choice.

However, nonlinear algorithms like neural nets require a vast amount of data for training. A lack of sufficient data may introduce instability, making simpler methods preferable to more complex ones. In the present paper, we show that for equity return predictions, a simpler method, which we christen VASA, provides similar and even superior results than some benchmark neural networks and random forests. VASA is a sobriquet for **va**riable **s**ubsample **a**ggregation.[2] As the benchmark for VASA, we use the methods used in Gu et al. (2020). VASA is a straightforward subsampler in the factor space, aggregating several linear models based on a specific weighting of each submodel.[3] For both the simulation as well as the empirical analysis, the performance of VASA turns out to be remarkably well. We attribute the predictive gains not only to a dimension reduction, by subsampling in the predictor space, but also to explicitly account for potential model-selection mistakes, by averaging over multiple subsampled factor models.

For our analysis, we depart from the setup of Gu et al. (2020) in at least three points. First,

[1] Over the recent decades, factors become so numerous that Cochrane (2011) referred to the collection as a zoo. See also, e.g., Harvey et al. (2016), Hou et al. (2018), and others.

[2] Vasa is also a retired Swedish warship that foundered after sailing about 1,300 meter, just outside the Stockholm harbor, into its maiden voyage in the 17th century. Vasa sank because it had very little initial stability due to the wrong estimation of the distribution of mass in the hull structure and the loading ballast. Our VASA aims to improve the predictive power of some estimator or algorithm by reducing the curse of dimensionality. Therefore, maybe the proposed VASA method could have prevented the Vasa sinking, by not putting too much weight too high in the ship, using an appropriate weighting function.

[3] Indeed, we can interpret VASA as a special case of a neural net with independent learning between the submodels and a subset of parameters, having a weight of zero by default. We also note that the wording subsample is in the literature mainly used in the context of taking subsets in the observation space, where each observation contains all the predictors. In our framework, a subsample refers to a subset in the predictor space, containing all observations.

we argue that to validate different methods, we must not only look at the global predictive out-of-sample R^2 (R^2_{OOS}). The R^2_{OOS} was suggested by Campbell and Thompson (2008) to evaluate the forecast accuracy of different models by measuring the proportional reduction in the mean-squared forecasting error for a candidate model relative to the benchmark.[4] We take the position that for the performance measurement of portfolio strategies using a large panel of stock return data, the R^2_{OOS}-measure is an insufficient measure. It may be misleading as it neglects the danger of outliers with extreme or false predictions. Therefore, we propose an individual or stock-specific $R^2_{OOS,i}$, which allows us to analyze the cross-sectional distribution of stock-specific forecasting accuracies. Only in this way can we thoroughly compare the prediction performance of the different machine learning algorithms.

Second, for our simulation exercise, we adopt the same hybrid sample splitting scheme as we do for the empirical analysis, i.e., we recursively increase the training sample, periodically refit the different models to make out-of-sample predictions over the subsequent year. Hence, we increase the training sample each period while maintaining a fixed size rolling sample for validation. By applying the same scheme to both simulation and empirical analysis, we can investigate whether our findings from the simulation translate consistently to the case with empirical data.

Third, for our empirical analysis, we only focus on a subset of the stocks considered in Gu et al. (2020). Our goal is to obtain balanced panel data. Furthermore, we want to obtain the distribution of $R^2_{OOS,i}$ over the whole sample period. Therefore, we only focus on the stocks for which the entire return history is available. Our sample period starts in January 1977 and ends in December 2016, totaling 40 years, which leaves us with a universe of 501 stocks. Gu et al. (2020) argue that machine learning methods "are most valuable for forecasting larger and more liquid stock returns and portfolios". Our sample of stock returns consists of stocks for which a full return history is available. Generally, these are liquid stocks of larger companies. Hence, we may expect that the machine learning methods explored in Gu et al. (2020) provide a conservative benchmark for VASA.[5]

[4] To calculate the R^2_{OOS}, the historical average model is usually defined as the relevant benchmark model. We follow this convention.

[5] Gu et al. (2020) include stocks with prices below \$5, share codes beyond 10 and 11, and financial firms. They end up with an average number of stocks per month exceeding 6,200. While using a larger sample helps to avoid overfitting, they argue that their results are qualitatively identical and quantitatively unchanged if they would filter out these firms. Therefore, we think that by restricting our dataset to $N = 501$, we do not lose generality.

In our simulation analysis, we find that VASA works remarkably well under different levels of sparsity and signal-to-noise ratios. While the best-performing method in the linear baseline simulation is LASSO, VASA performs at least as good or better than LASSO for other levels of sparsity. Moreover, similar to penalized linear methods, VASA shows a high prediction accuracy and low variability across different signal-to-noise ratios, while the nonlinear methods stay behind. However, when we increase the signal-to-noise ratio, the neural network starts to perform well. Clearly, when we use a nonlinear data-generating process, the nonlinear methods, especially random forests, provide the best performance. However, VASA's performance remains robust, providing a higher predictive R^2 than, e.g., the neural network and the penalized linear models. Hence, in the simulation exercise, we find that VASA performs remarkably well across different variations of the data generating process. The robustness of these results gives us hope that they also carry over to the empirical analysis.

When we take the different methods to the data, it becomes evident that the global R^2_{OOS} might not be a sufficient measure to guarantee superior portfolio performance. The distribution of the individual $R^2_{OOS,i}$ plays a crucial role. We find that outliers matter for building long-short portfolios from return predictions. For stock return predictions using a set of 94 predictors, VASA provides the highest global R^2_{OOS}. When we increase the predictors to 904, we find that random forests offer the highest global R^2_{OOS}. However, the standard deviations of the $R^2_{OOS,i}$ is almost four times larger than for VASA. This high variability has a detrimental impact on the performance of the long-short portfolio based on the random forest's return prediction. Although VASA's global R^2_{OOS} is lower, the resulting Sharpe ratio is 73% larger than the one using the random forest. Indeed, due to the low variability in VASA's $R^2_{OOS,i}$'s, the portfolio strategy built on VASA consistently and markedly outperforms all other portfolios.

While VASA is a new method, it is inherently related to at least two streams of literature. First, VASA is closely related to three common regularization techniques, RIDGE, LASSO, and dropout. VASA is similar to LASSO in that we do model-selection via subsampling in the predictor space, but with the advantage that it controls for potential model-selection bias by aggregating different (probability-weighted) submodel predictions. Besides, VASA is close to dropout regressions introduced by Srivastava et al. (2014a), where we set at random some of the elements in the design matrix to zero such that any input dimension is retained. As shown in

Srivastava et al. (2014a, Section 9.1), under some additional assumptions dropout with linear regression is equivalent to RIDGE regression. While VASA is related to RIDGE and LASSO, it has the advantage that the underlying model does not need to be linear. Hence, we could also incorporate nonlinear models into the VASA setting. Such considerations will be of interest for future research. For now, we focus on VASA with linear models.

To some extent, VASA is also related to the literature on ensemble methods such as random forests. Ensembles are sets of learning machines that combine in some way their decisions, or their learning algorithms, or different views of data, or other specific characteristics to obtain more reliable and more accurate predictions in supervised and unsupervised learning problems. See, e.g., Dietterich (2000a). Empirical studies showed that both in classification and regression problems ensembles improve on single learning machines.[6] Recently, Jacobsen et al. (2019) introduce ensemble machine learning for stock return prediction. The average forecasts from different linear models, namely Bayesian model averaging, LASSO, and weighted least-square, based on random subsamples and adaptively changing the sampling distribution. Rossi (2018) follows a similar approach, but uses nonlinear models. Our VASA substantially differs from these models in that we do not subsample from the observation space. Instead, we select a subsample from the predictor space and aggregate the resulting predictions.

The literature on the application of machine learning methods in finance has seen a significant surge in articles related to the prediction of the cross-section of stock returns. For example, Moritz and Zimmermann (2016) use random forests to predict stock returns based on 86 firm characteristics.[7] Freyberger et al. (2020) estimate expected stock returns based on a set of 62 characteristics using adaptive group LASSO. Gu et al. (2020) compare a wide variety of different machine learning methods, ranging from penalized linear models to random forests and neural nets.[8] Chen et al. (2019) improve on the methods of Gu et al. (2020) by integrating no-arbitrage conditions into different machine learning algorithms. Rasekhschaffe and Jones (2019) show that an ensemble of different machine learning methods improves return predictions. All of the above literature concludes that more complex machine learning methods beat simple linear models for cross-sectional return prediction based on a multitude of factors.

[6] See, e.g., Bauer and Kohavi (1999); Dietterich (2000b); Banfield et al. (2006).
[7] See also, e.g., Coqueret and Guida (2018).
[8] The best performing model of Gu et al. (2020) has become a benchmark for many other papers on machine learning in finance, e.g., Chen et al. (2019).

We organize the remainder of the paper as follows. Section 2 gives a short overview of the benchmark models used in our study and provides the details on our new subsampling framework. Section 3 examines and compares the finite-sample behavior of the different methods via Monte Carlo simulations. In Section 4, we describe the empirical methodology and present the results of the out-of-sample backtest exercise based on historical stock returns and stock characteristics. Section 5 concludes.

2 Methodology

We consider N assets and denote the (excess) return of asset i over one period from t to $t+1$ by $r_{i,t+1}$ with $i \in \{1, \ldots, N\}$ and $t \in \{1, \ldots, T\}$. In its most general form, we describe an asset (excess) return $r_{i,t+1}$ as an additive prediction error model:

$$r_{i,t+1} := \mathbb{E}_t[r_{i,t+1}] + \epsilon_{i,t+1} \ , \tag{2.1}$$

where we assume the conditional expectation $\mathbb{E}_t[r_{i,t+1}]$ to be a function of a set of predictors, i.e.,

$$\mathbb{E}_t[r_{i,t+1}] := g(\boldsymbol{z}_{i,t}) \ , \tag{2.2}$$

where $\boldsymbol{z}_{i,t} := (z_{i,t,1}, \ldots, z_{i,t,P})'$ is a vector of P predictors. For our analysis, $\boldsymbol{z}_{i,t}$ is a mixture of asset specific factors (characteristics) and macroeconomic variables. The function $g(\cdot)$ is a flexible function of these predictors. Even with the assumed structural form from Equation (2.1), there are infinitely many ways to define and estimate the function $g(\cdot)$.

Instead of focusing on an asset-specific functional form of $g_i(\cdot)$, we assume a panel-data model that maintains the same functional form over time and across different assets, i.e., $g_i(\cdot) \equiv g(\cdot)$ for all i; see Rosenberg (1974), Harvey and Ferson (1999), Gu et al. (2020), among others.[9] It is an interesting question in itself whether a panel-data model improves upon asset-specific pricing models by leveraging information from the entire panel, which lends stability to estimates of risk premia for any individual asset. When we implemented an asset-specific approach for

[9] To be precise, while the simulation analysis in Gu et al. (2020) is indeed a panel-data model, their empirical analysis is not since the stock universe changes during the sample period. Hence, they call it an "over-arching approach". For our analysis, and to make the simulation analysis consistent with our empirical investigation, we only select stocks for which the whole history of returns is available. Hence, we use in both cases a panel-data approach.

our empirical analysis, we found that the results were disappointing across all methods.[10] We conjecture that the asset-specific approach fails in the current setting due to the small sample size and low-frequency of a large number of characteristics.[11] Therefore, we do not present these results here.

2.1 Benchmark Methods

We first provide a short overview of the linear and nonlinear methods, which we use as a benchmark for VASA, our subsampled factor model. Of course, one could explore many more methods, but we wanted to have a set of methods that is close to the benchmark analysis of Gu et al. (2020).

2.1.1 OLS

We start with the least complex method in our analysis, the simple linear predictive regression model estimated via ordinary least squares (OLS). While we expect OLS to perform poorly in our high-dimension setting, we use it as a reference point for emphasizing the distinctive features of more advanced methods. OLS assumes that the conditional expectation of the asset returns can be approximated by a simple linear function of the predictor vector

$$r_{i,t+1} = \alpha + \boldsymbol{z}_{i,t}'\boldsymbol{\beta} + \epsilon_{i,t+1} \ , \tag{2.3}$$

where α denotes the intercept and $\boldsymbol{\beta}$ is the P-dimensional coefficient vector. Then, OLS solves the following optimization problem

$$\min_{a,\boldsymbol{b}} \sum_{i=1}^{N}\sum_{t=1}^{T}(r_{i,t+1} - a - \boldsymbol{z}_{i,t}'\boldsymbol{b})^2 \ , \tag{2.4}$$

with the solution $\hat{\alpha}_i$ and $\hat{\boldsymbol{\beta}} := (\hat{\beta}_1, \ldots, \hat{\beta}_P)'$. However, for large-dimensional factor models, where P is of similar order than $N \times T$, the high variability in the least squares fit results in overfitting

[10] These results are not reported here, but they can be obtained on request.

[11] A large number of characteristics leads to a large-dimensional problem with (effective) concentration ratio (above) close to 1. In the empirical analysis, when using an asset-specific approach, the effective concentration ratio is much larger than $\mathcal{C} = P/T = 94/144 \approx 0.65$ as most of the characteristics are quarterly or even annually. Thus the effective sample size is $T^{\text{effective}} = (20 \times 144 + 13 \times 36 + 61 \times 12)/144 \approx 28$ and not 144, leading to a problematically high effective concentration ratio $\mathcal{C}^{\text{effective}} = P/T^{\text{effective}} \approx 3.32$.

and consequently poor predictions for out-of-sample observations not used in model training. Additionally, if the amount of covariates P is larger than the number of observations $N \times T$, the minimization problem above is not solvable.

As a special case, we can assume that we have no covariates at all by setting $\boldsymbol{\beta} = (0, \ldots, 0)'$. Then,

$$\hat{r}_{i,T+1} = \hat{\alpha} = \frac{1}{N \times T} \sum_{i=1}^{N} \sum_{t=1}^{T} r_{i,t}, \tag{2.5}$$

i.e., the predictor is equal to the average.

2.1.2 RIDGE and LASSO

The simple linear model is bound to fail in the presence of many predictors. Especially for a very low signal-to-noise ratio, it begins to overfit noise rather than extracting signal. Arguably, RIDGE and LASSO regressions are the standard machine learning methods to 'reduce' the number of estimated parameters, and thus avoiding overfitting. The goal of these penalized linear methods is to improve out-of-sample predictions stability by shrinking the regression coefficients.

RIDGE shrinks the regression coefficients towards zero by imposing an L_2 penalty on their size. The minimization becomes

$$\min_{a,\boldsymbol{b}} \sum_{i=1}^{N} \sum_{t=1}^{T} (r_{i,t+1} - a - \boldsymbol{z}'_{i,t}\boldsymbol{b})^2 + \lambda \sum_{p=1}^{P} b_p^2 \,, \tag{2.6}$$

where λ is a regularization parameter between 0 and infinity. In the special case with $\lambda = 0$, we obtain the basic OLS solution of Equation (2.4) and $\lambda = \infty$ returns the intercept-only model of Equation (2.5).

LASSO includes a L_1 penalization that makes the solution nonlinear without closed-form expression. LASSO is a variable selection method that imposes sparsity on the specification and sets coefficients on a subset of covariates exactly to zero (least absolute shrinkage and selection operator). The minimization becomes

$$\min_{a,\boldsymbol{b}} \frac{1}{2} \sum_{i=1}^{N} \sum_{t=1}^{T} (r_{i,t+1} - a - \boldsymbol{z}'_{i,t}\boldsymbol{b})^2 + \lambda \sum_{p=1}^{P} |b_p| \,. \tag{2.7}$$

In our empirical analysis we do not consider the use of elastic nets, which is a convex combination of the two penalization methods.[12]

2.1.3 Random Forest

A traditional random forest (RF), as introduced by Breiman (2001), is a collection of trees, where each tree is trained using a bootstrap sample of the original data. Each tree only uses a subset of the observations for training, which is an essential source of randomness. A single regression tree partitions the P dimensional predictor space into M subspaces according to a set of splitting rules. Each created partition contains a particular bag of observations. The estimator is the average over the responses in the respective subset.

The tree structure, or partition, is built by iteratively splitting the predictor space into rectangular subspaces. At each node of the tree a variable and a splitpoint is chosen, such that the sum of squared in-sample prediction errors are minimized. The quality of a partition $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ – created by a certain variable and splitpoint – can be assessed by the objective function

$$\sum_{i,t \in \mathcal{P}_1} (r_{i,t+1} - \bar{r}^{(1)})^2 + \sum_{i,t \in \mathcal{P}_2} (r_{i,t+1} - \bar{r}^{(2)})^2 \ ,$$

where $\bar{r}^{(j)} = \frac{1}{|\mathcal{P}_j|} \sum_{i,t \in \mathcal{P}_j} r_{i,t+1}$ denotes the average of partition j. At each splitpoint an observation can either go left or right depending on the splitting criterion. Note that a random forest uses only a subset of the predictors at each node to generate a split. This, in addition to the bootstrap sample, reduces the correlation between the trees even further.

For B trained regression trees and an arbitrary observation $\boldsymbol{z}_{i,t}$, the random forest takes the average over the B individual tree predictions. In that sense, a random forest can be seen as a local average estimator (Devroye et al., 2013, Section 6.5). Typically, a random forest tree is fully grown and unpruned, such that each tree has a minimal bias, but a larger variance.[13] Each tree would be prone to overfitting. However, the average reduces the variance by keeping the unbiasedness.

We remark that, as of today, it has not been proven that a random forest, as introduced in

[12] We use the R package "glmnet" by Friedman et al. (2010) for both the LASSO and the RIDGE.

[13] In this reasoning, each tree is seen as a random object reflecting the necessary information from the splits of the feature space into smaller regions.

this section, is consistent for the true underlying asset specific regression function g_i.[14] There are cases in which random forests fail, for certain setups it is even possible to show inconsistency; see Tang et al. (2018). However, from a practical viewpoint it is well known that random forests perform well in prediction problems. Moreover, in contrast to neural networks, the prediction accuracy of random forest is less sensitive to the tuning parameters.[15] Especially the number of trees does not necessarily need to be chosen via cross-validation (Hastie et al., 2009, p.596).[16]

2.1.4 Neural Net

The most common type of neural network (NNET) model is the feed-forward multi-layer perceptron. Gu et al. (2020) argue that shallow learning outperforms deeper learning. Hence, we only present the results for a shallow neural network with four hidden layers. This choice corresponds to their best performing network.[17]

The elements of a basic neural network are an input layer with P elements (each variable is a knot), the hidden layers, and a linear output. The hidden layers have 32, 16, 8, and 4 neurons, respectively. Finally, all the layers are fully connected via the weight parameters. Hence, the return of asset i at time $t+1$ is given by

$$r_{i,t+1} = \alpha_1 + W_1\phi(\boldsymbol{\alpha}_2 + W_2\phi(\boldsymbol{\alpha}_3 + W_3\phi(\boldsymbol{\alpha}_4 + W_4\phi(\boldsymbol{\alpha}_5 + W_5\boldsymbol{z}_{i,t})))) + \epsilon_{i,t+1} \ , \tag{2.8}$$

where the activation function, ϕ is applied elementwise and $\{\alpha_1, \ldots, \boldsymbol{\alpha}_5, W_1, \ldots, W_5\}$ is the set of biases and weight matrices.[18]

As activation function for all neurons in all hidden layers, we choose the commonly used Rectifier linear unit (Hahnloser et al., 2000):

$$\phi(x) = \text{ReLU}(x) = \max(0, x) \ . \tag{2.9}$$

Our architecture has a total of $(32+32\times P)+(16+16\times 32)+(8+8\times 16)+(4+4\times 8)+(1+1\times 8) =$

[14] By consistency, we mean consistency in ℓ_2-norm: $\lim_{N\to\infty} \mathbb{E}[(\hat{g}(\boldsymbol{z}_{i,t}) - g(\boldsymbol{z}_{i,t}))^2]$.
[15] See, e.g., Liu et al. (2013), Ahmad et al. (2017).
[16] For the random forest, we use the R package "ranger" by Wright and Ziegler (2017) with 500 trees, max.depth $\in \{0, 1, ..., 6\}$ and mtry $\in \{3, 5, 10, 20, 30, 50\}$.
[17] We also analyzed a network with only one hidden layer. The results were disappointing and, hence, we do not report them here.
[18] The final output layer projects on only one neuron, hence W_1 is a vector of size four.

$741 + 32 \times P$ parameters to estimate.

We implement the architecture as described above in TensorFlow (Abadi et al., 2015). As in Gu et al. (2020), we used the Adam optimization algorithm (Kingma and Ba, 2014), early stopping, batch normalization (Ioffe and Szegedy, 2015), and ensembles. However, instead of weight decay, we use dropout in each hidden layer (Srivastava et al., 2014b) as additional prevention of overfitting.[19]

2.2 Variable Subsampling Aggregation (VASA)

As an alternative to the least absolute shrinkage and selection operator like LASSO and RIDGE as outlined above, there exists a large strand of literature on (best) subset selection methods for linear regressions.[20] By retaining a subset of the predictors and discarding the rest, subset selection may improve the model's interpretability. However, subset selection methods in linear regression often perform poorly in terms of variable selection, estimation of coefficients, and standard errors — especially in situations that are typical for finance: a large number of (different) variables and the presence of multicollinearity.

To overcome these deficiencies, we propose VASA as a subsampling procedure that does not suffer from high variability and model-selection bias by averaging over multiple subsampled (factor) model predictions.[21] The effectiveness of averaging over several bagged estimators to reduce variance is undisputed in machine learning. However, for linear models, the benefit is controversial, especially when all assumptions are met. Notable is that instead of taking a bootstrap sample in the observational space, we suggest taking a subset in the predictor space.

The general procedure for VASA is based on the idea that the final estimator (predictor) is

[19] We train the neural network via 100 epochs with batch sizes of 10,000 observations. These results do not differ much when choosing a smaller batch size. In the Adam stochastic gradient descent algorithm, we use an initial learning rate of 0.01, and we leave the other parameters at their default value. Further, we choose the dropout probability via the validation set from a set of 19 values between 0.175 and 0.625. Finally, we set the number of ensembles equal to 10. For a more detailed explanation, some good references for neural networks of this form are Bishop et al. (1995), Hertz et al. (1991), Ripley (1993), Ripley and Hjort (1996), Friedman et al. (2001) and Bishop (2006).

[20] See e.g. Hastie et al. (2009, Chapter 3.3). The idea behind subset selection is that we retain only a subset of the predictors and eliminate the rest from the model. Different approaches exist to choose this subset, such as best-subset selection, forward-stagewise regression, forward- and backward-stepwise selection. See, e.g., Hastie et al. (2017) for a comparison. Lastly, to estimate the coefficients of the inputs that are retained, most often least squares regression is used.

[21] Note that VASA is a general subsampling framework and model-independent. In this paper, we present only a simple application to large-dimensional factor models. However, VASA is not restricted to a simple linear regression and can be applied to any base algorithm.

a combination of base estimators, where we apply each base algorithm only to a subset of the variables. Hence, the main intuition behind VASA is the fact that each base algorithm is only confronted with a subset of all the features. This feature can be game-changing when the amount of variables is at the same magnitude or higher than the number of observations. Further, this fact offers a great opportunity in variable selection — depending on how the weights are chosen. Therefore, VASA is a dimension reduction method that counters the curse of dimensionality.

The last step for VASA consists in performing an aggregation by a function that weighs the individual algorithms in a certain way. A naive approach would be taking the average. However, in the case of regression, we may rely on alternative ways to find an "optimal" aggregation function by choosing the weights according to some loss function. In this context, the literature on boosting offers many possible solutions; see, for example, Bühlmann et al. (2007).[22] Hence, to sum up, VASA reduces the curse of dimensionality by drawing subsets based on some variable importance measure and additionally reduces the variability and model-selection bias by aggregating their predictions.

2.2.1 VASA in General

VASA can be applied to a variety of different settings. Here, our focus is on the application of VASA in a regression setting. As before, our data consist of pairs $(\boldsymbol{z}_{i,t}, r_{i,t+1})$ for $i = 1, \ldots, N$ and $t = 1, \ldots, T$, where $\boldsymbol{z}_{i,t} \in \mathbb{R}^P$ denotes the P-dimensional predictor variable at time t to perform an out-of-sample prediction for the next period's return $r_{i,t+1} \in \mathbb{R}$. Recall, our target function is the conditional expectation $\mathbb{E}_t[r_{i,t+1}]$. Given the stock specific characteristics and the returns, VASA trains B submodels via a common base algorithm (for example OLS). For the training of each submodel only a subset of the P variables are used. Finally, for an arbitrary input characteristic vector $\boldsymbol{z}_{i,t}$, the proposed VASA predictor, $\hat{g}^{\text{VASA}}$, is the aggregation of B subpredictions $\hat{g}^{BASE}(\tilde{\boldsymbol{z}}_{i,t,1}), \ldots, \hat{g}^{BASE}(\tilde{\boldsymbol{z}}_{i,t,B})$ via an aggregation function $f : \mathbb{R}^B \to \mathbb{R}$. Each subprediction originates from a base algorithm trained only on a subsample in the predictor space, where $\tilde{\boldsymbol{z}}_{i,t,b}$ is the b-th input vector of size $K_b \leq P$ (the subsample size) containing only the variables which were used for training the b-th submodel. To summarize, we predict $r_{i,t+1}$

[22] The concept of VASA relates, at least in some aspects, to the concept of a super learner, proposed by Van der Laan et al. (2007). However, unlike in VASA, a super learner aggregates different learning methods, each using all the variables in the feature space.

as follows,

$$\hat{r}_{i,t+1} = \hat{g}^{\text{VASA}}(\boldsymbol{z}_{i,t}) := f\left[\hat{g}^{BASE}(\tilde{\boldsymbol{z}}_{i,t,1}), \ldots, \hat{g}^{BASE}(\tilde{\boldsymbol{z}}_{i,t,B})\right]. \tag{2.10}$$

The procedure is described in Algorithm 1. Without loss of generality, we can rewrite the aggregation function, f, as a sum of weights such that the VASA predictor becomes

$$\hat{g}^{\text{VASA}}(\boldsymbol{z}_{i,t}) := \sum_{b=1}^{B} \omega_b \hat{g}^{BASE}(\tilde{\boldsymbol{z}}_{i,t,b}) \ , \tag{2.11}$$

where ω_b is the weight of the b-th subsample satisfying the two conditions $\sum_{b=1}^{B} \omega_b = 1$ and $0 \leq \omega_b \leq 1 \ \forall \ b = 1, \ldots, B$. Moreover, if we set $\omega_b = 1$, we consider only the b-th subsample for the estimation. For example, $\omega_b = \frac{1}{B}$ and $K_b = \frac{p}{2}$ for $b = 1, \ldots, B$ would be B-times half-sampling with equal weights. Thus, VASA is a generalization of the classical regression method, if we set $B = 1$ and $K_b = P$. For each time point we can write the randomly generated subsamples $\tilde{\boldsymbol{z}}_{t,b}$ as

$$\tilde{\boldsymbol{z}}_{t,b} := \tilde{\boldsymbol{z}}_t \Lambda(V_b)' \ , \tag{2.12}$$

where $\tilde{\boldsymbol{z}}_{t,b} \in \mathbb{R}^{N \times b}$ and $\Lambda(V_b) \in \{0,1\}^{K_b \times P}$ is the variable selection matrix based on the P-dimensional subsampling vector $V_b \in \{0,1\}^P$. Hence, for $V_b = \{v_{b,1}, \ldots, v_{b,P}\}'$ such that $V_b' \mathbb{1} = K_b$, where $\mathbb{1}$ denotes a vector of ones of dimension $P \times 1$; $v_{b,l} = 1$ indicates that the variable $l \in \{1, \ldots, P\}$ is selected, whereas $v_{b,l} = 0$ indicates that the variable l is not selected. In Appendix A, we illustrate the selection matrix $\Lambda(V_b)$ and we prove that, under the assumption that the N pairs $(\boldsymbol{z}_{i,t}, r_{i,t+1})$ are i.i.d., the subsampling vector V_b is distributed according to a special case of the multivariate hypergeometric distribution. Hence, we can write the VASA matrix, which identifies the randomly generated subsamples, as

$$V := \{V_1, \ldots, V_b, \ldots, V_B\} \quad \in \{0,1\}^{P \times B} \ , \tag{2.13}$$

with $V \mathbb{1}_B$ denoting the frequency vector containing the number of selections of each variable.

We remark that we can generalize VASA by subsampling the feature space according to a given discrete probability distribution $\mathbb{P}$. The proposed VASA estimator from Equation (2.10) and Algorithm 1, with randomly generated subsamples and thus $V_b \sim \text{HGeom}(p, K_b)$, is just a special case where we set $\mathbb{P} \sim \mathcal{U}\{1, p\}$. If any additional information about the relevance of the

Algorithm 1 VASA $\leftarrow$ function$(x, \boldsymbol{z}, r, B, K_b, f)$

Require: $x \in \mathbb{R}^{1\times P}$, $\boldsymbol{z} \in \mathbb{R}^{N\times T\times P}$, $r \in \mathbb{R}^{N\times T}$, $B \in \mathbb{N}$, $K_b \in \{1,\dots,P\}\ \forall\ b = 1,\dots,B$ and an aggregation function $f : \mathbb{R}^B \to \mathbb{R}$

1: **for** b in $1:B$ **do**
2: $V_b \leftarrow$ randsample$(p, K_b,$ replacement=FALSE$)$ // $V_b \sim$ HGeom(P, K_b)
3: $\tilde{\boldsymbol{z}}_b \leftarrow \tilde{\boldsymbol{z}}[:,:,V_b]$
4: Estimate the b-th submodel using $\tilde{\boldsymbol{z}}_b$ and the response r
5: **return** $f\left[\hat{g}^{BASE}(x[V_1]),\dots,\hat{g}^{BASE}(x[V_1])\right]$

variables or features for the parameter estimation were available, we could easily deviate from the discrete uniform distribution.

2.2.2 VASA with Linear Submodels

So far, we did not specify each base predictor $\hat{g}^{BASE}$ in Equation (2.10). Since we want to compare our method mainly with linear methods, we assume that our target function, the conditional expected return, is a linear function of the predictor variables. Under this assumption, we create the VASA prediction as an average of B OLS-predictions, each trained on a (pseudo) random subset of the P predictors, i.e.,

$$\hat{g}^{\text{VASA}}(\boldsymbol{z}_{i,t}) := \sum_{b=1}^{B} \omega_b \hat{g}^{\text{OLS}}(\tilde{\boldsymbol{z}}_{i,t,b}) = \sum_{b=1}^{B} \omega_b (\hat{\alpha}_b + \tilde{\boldsymbol{z}}'_{i,t,b}\hat{\boldsymbol{\beta}}_b)\ , \tag{2.14}$$

where $\omega_b \in [0,1]$ is the weight of the b-th OLS prediction with $\sum_{b=1}^{B}\omega_b = 1$ and $\hat{\boldsymbol{\beta}}_b$ is a K_b-dimensional vector, where K_b represents the dimension (subsample size) of submodel b. For ease of interpretability, we assume that the optimal subsampling size is constant across the submodels, thus $K \equiv K_b$. Hence, each $\tilde{\boldsymbol{z}}_{i,t,b}$ contains K (pseudo) randomly chosen variables (without replacement) from the P predictors.

The number of submodels B and their dimension K are tuning parameters. For each submodel $b = 1,\dots,B$, the estimation problem is analogous to Equation (2.4), i.e.,

$$\min_{\alpha_b,\boldsymbol{b}_b} \sum_{i=1}^{N}\sum_{t=1}^{T}(r_{i,t+1} - \alpha_b - \tilde{\boldsymbol{z}}'_{i,t,b}\boldsymbol{b}_b)^2\ , \tag{2.15}$$

with the small but important difference that $\tilde{\boldsymbol{z}}_{i,t,b}$ does not contain all predictor variables. The most basic model is to take equally distributed subsampling probabilities, $q_p = 1/P$ and weights

$\omega_b = 1/B$. However, we calculate the subsampling probabilities proportional to the P univariate regression R^2

$$q_p = R^2_{i,p} / \sum_{j=1}^{P} R^2_{i,j} \ ,$$

where $R^2_{i,p}$ is the in-sample R^2 from regressing r_i on the p-th factor. The intuition behind our choice of the subsampling probabilities is that a variable which seems to be (more) relevant for explaining asset returns, at least in-sample, should be included more frequently in a submodel and vice versa.[23]

Alternatively, one could use the variable importance measure from random forest or others to infer reasonable subsampling probabilities.

2.3 Sample Splitting and Performance Evaluation

For the performance evaluation and model comparison, we follow the standard design of disjoint subsamples for estimation, validation, and testing that maintains the temporal ordering of the data:

1. A *training* sample, comprising of the first 30% observations. We use it to estimate the parameters of our models subject to some initial specification for its hyperparameters.

2. A *validation* sample, retaining the successive 20% of observations. This sample allows us to optimize the hyperparameters of our models directly from next year's data. The hyperparameters are critical to the performance of machine learning methods as they control model complexity and thus overfitting. For our models, the hyperparameters include the penalization parameter λ for RIDGE and LASSO, the number of subsampled factor models B and their dimension K for VASA, the number of variables at each split point to choose from and the depth of the trees in a random forest, and finally the dropout probability in the neural net.

3. A *testing* sample containing the next (last) twelve months of data. These data, which

[23] Given this specification, VASA is similar to LASSO by doing model-selection via subsampling in the predictor space, but with the advantage that it controls for potential model-selection bias by aggregating B submodel predictions. In addition, VASA is close to dropout regression by Srivastava et al. (2014a), where at random some of the elements in the design matrix Z_i are set to zero such that any input dimension is retained. A nice link between dropout regression and RIDGE can be found in Srivastava et al. (2014a, Section 9.1).

never enter the parameter optimization procedure, are used to test the predictive capability of our models.

In our empirical exercise and simulation study, we adopt a hybrid sample splitting schemes similar as in Gu et al. (2020), i.e., we recursively increase the training sample and refit the entire model once per year. With the fitted model, we make return predictions over the subsequent year. While we grow the training sample by a year whenever we refit the model, we maintain a fixed-size rolling sample for validation. We choose not to cross-validate to preserve the temporal ordering of the data for prediction. To compare the performance of various models, we look at the *asset specific* out-of-sample $R^2_{OOS,i}$

$$R^2_{OOS,i} := 1 - \frac{\sum_{t\in\mathcal{T}}(r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{t\in\mathcal{T}} r^2_{i,t+1}} \ , \tag{2.16}$$

where $\mathcal{T}$ indicates that the model predictions where only assessed on the testing sample.

A subtle but important aspect of our methodology is that we independently use an asset-specific performance measure, which is more informative as it allows us to obtain some information about the cross-sectional distribution of stock-specific forecast accuracies. Arguably, it is of interest to see the prediction accuracy for each asset to avoid extreme and thus often unrealistic predictions. Indeed, in our empirical analysis, we observe that outliers matter and should be taken into account for the performance evaluation of the models. Hence, we are interested in the cross-sectional distribution of the out-of-sample $R^2_{OOS,i}$. In particular, we report four out-of-sample performance measures for each scenario, the median, the average, the standard deviation, and the $10th$ percentile (decile) of the N out-of-sample $R^2_{OOS,i}$.

3 Monte Carlo Simulation

Before we analyze real data, we run an extensive Monte Carlo simulation to evaluate the prediction accuracy of the different models introduced in the previous section. To robustify our results, we analyze various changes in the input parameters and data generating process.

3.1 Data Generating Process

For the specification of the data generating process, we follow the approach of Gu et al. (2020). For completeness, we briefly describe the setup. To generate the P_c stock characteristics $c_{i,p,t}$ and returns $r_{i,t}$, for $i = 1, \ldots, N$, $p = 1, \ldots, P_c$ and $t = 1, \ldots, T$, we define an AR(1) process for an auxiliary variable $\bar{c}_{i,p,t}$,

$$\bar{c}_{i,p,t} := \rho_p \bar{c}_{i,p,t-1} + e_{i,p,t} \ , \tag{3.1}$$

where $\bar{c}_{i,p,0} = 0$, $\rho_p \sim \mathcal{U}_{[0.9,1]}$ and $e_{i,p,t} \sim \mathcal{N}(0, 1 - \rho_p^2)$. We then use this auxiliary variable to generate the cross section and time series of all the characteristics

$$c_{i,p,t} := \frac{2}{N+1} \text{CSRank}(\bar{c}_{i,p,t}) - 1 \ , \tag{3.2}$$

where CSRank$(\cdot)$ is the cross-section rank function. Hence, the resulting characteristics features will exhibit some degree of persistence over time.

In addition to the stock characteristics, we simulate a time series x_t representing the (macro-) economic environment. We base x_t on the following model:

$$x_t := \rho x_{t-1} + u_t \ , \tag{3.3}$$

where $x_0 = 0$, $\rho = 0.95$ and $u_t \sim \mathcal{N}(0, 1 - \rho^2)$. Hence, x_t is highly persistent. The macroeconomic variable x_t enters our data generating process through a Kronecker product

$$\boldsymbol{z}_{i,t} := (1, x_t)' \otimes \boldsymbol{c}_{i,t} \ , \tag{3.4}$$

where $\boldsymbol{c}_{i,t}$ is the P_c-dimensional stock characteristics vector at time t. Thus, we add P_c interaction terms between the stock characteristics and the macroeconomic variable, which results in a $(P = 2P_c)$-dimensional covariate vector $\boldsymbol{z}_{i,t}$. Before we can use the large-dimensional covariate vector $g(\boldsymbol{z}_{i,t})$ to predict stock returns $\hat{r}_{i,t+1}$, we define a latent K^*-factor model to generate (excess) returns

$$r_{i,t+1} := g^*(\boldsymbol{z}_{i,t}) + \epsilon_{i,t+1} \ , \tag{3.5}$$

with

$$\epsilon_{i,t+1} := \boldsymbol{v}'_{t+1}\boldsymbol{\beta}^*_{i,t} + \varepsilon_{i,t+1} \ , \tag{3.6}$$

where $\boldsymbol{v}_{t+1} \sim \mathcal{N}(\mathbf{0}, 0.05^2 \times \mathbb{I}_3)$ represents a K^*-dimensional disturbance vector for the stocks drawn from the multivariate normal distribution and heavy tailed idiosyncratic errors $\varepsilon_{i,t+1} \sim t_5(0, 0.05^2)$.

As in Gu et al. (2020), we suggest to introduce sparsity by simulating a three-factor model with $\boldsymbol{\beta}^*_{i,t} = (c_{i,1,t}, c_{i,2,t}, c_{i,3,t})'$ and we use two cases for the functional form $g^*(\boldsymbol{z}_{i,t})$:

Case 1:

$$g^*(\boldsymbol{z}_{i,t}) := (c_{i,1,t}, c_{i,2,t}, c_{i,3,t} \times x_t)\theta_0 = (\boldsymbol{\beta}^*_{i,t} \circ (1, 1, x_t)')'\theta_0, \tag{3.7}$$

where $\theta_0 = (0.02, 0.02, 0.02)'$.

Case 2:

$$g^*(\boldsymbol{z}_{i,t}) := (c^2_{i,1,t}, c_{i,1,t} \times c_{i,2,t}, \text{sgn}(c_{i,3,t} \times x_t))\theta_0, \tag{3.8}$$

where $\theta_0 = (0.04, 0.03, 0.012)'$.

Hence, the first model is linear and sparse, whereas the second model is highly nonlinear as it involves a squared term $c^2_{i,1,t}$, an interaction term $c_{i,1,t} \times c_{i,2,t}$, and a discrete variable $\text{sgn}(c_{i,3,t} \times x_t)$.

3.2 Simulation Design

The following choices turn out to be central for the simulation results, the form of the (true) population generating asset return function $g^*(\boldsymbol{z}_{i,t})$, the number of driving covariates K^*, the regression coefficient θ_0, and the data set dimension (N, T, P_c) with concentration ratio $\mathcal{C} := 2P_c/T = P/T$. Similarly as in Gu et al. (2020), we define the base-case scenario as follows:

(a) $g^*(\boldsymbol{z}_{i,t}) = (c_{i,1,t}, c_{i,2,t}, c_{i,3,t} \times x_t)\theta_0$

(b) $K^* = 3$

(c) $\theta_0 = (0.02, 0.02, 0.02)'$

(d) $N = 100$, $T = 480$, $P_c = 100$

We first run the simulations under the base-case scenario. Then, we run different sets of simulations, allowing for some deviations from the base-case values, which enables us to study the influence of the different parameters separately and to assess the performance of the methods for various market conditions and economic cycles.

As described in Section 2.3, the Monte Carlo simulation follows the hybrid sample splitting and performance evaluation, where we use the first 12 years of observations to train, the successive eight years to validate, and the subsequent twelve months to test the models. Therefore, we have for each stock 20 years of monthly out-of-sample return predictions that we can analyze with the mentioned predictive $R^2_{OOS,i}$ measure. In that sense, we use the same procedure as in our empirical section.

As in Gu et al. (2020), we additionally include an "Oracle" estimator for comparison. This estimator is a feasible estimator using (only) the true covariates and OLS regression form, given the data generating process $g^*(\boldsymbol{z}_{i,t})$. However, it is not necessarily the truly best estimator one can construct as the simulated returns also have a non-random error term, see Equation (3.6), which is why we put "Oracle" inside quotation marks.

3.3 Base-Case Scenario

We report the results of the base-case scenario in Table 1. All the methods perform similarly under the base-case scenario. Even an ordinary least-squares regression seems sufficient. However, the penalized linear models 'successfully' reduce the curse of dimensionality and significantly improve prediction accuracy. The same holds for VASA with comparable prediction performance to LASSO and RIDGE. Note that in our more realistic simulation setting VASA and the penalized linear models even challenge the "Oracle" estimator.

[Table 1 about here.]

The implemented NNET is detecting some signal compared to the intercept-only regression. However, the NNET is performing rather poorly compared to RF. We note that our NNET, which uses four hidden layers and whose choice is motivated by the findings in Gu et al. (2020), is only one possible implementation of a neural net. One would possibly find a neural net that performs similar to the other methods in this scenario – or even better.

To calculate the global R^2_{OOS}, we take as a benchmark the prediction performance of the intercept-only method, which takes the cross-sectional average as a predictor for each asset. Thus, due to the cross-section rank function this value is just 0; see Equation (3.2). However, when looking at the distribution of the asset-specific performance, the average does retrieve some of the signals in the data. It is important to emphasize that in this simulated base-case scenario, the global R^2_{OOS} is a sufficient measure to compare the different algorithms as it is consistent with the individual $R^2_{OOS,i}$ performance. However, when we leave this base-case scenario and, in particular, for historical data, this consistency will break up.

3.4 Nonlinearity

Next, we investigate the effect of the functional form of the data generating process. Most of the factor model literature assumes $g^*(\boldsymbol{z}_{i,t})$ to be linear, but arguably this is a strong assumption. Therefore, we consider the nonlinear model in (3.8) and compare the prediction accuracy of the different methods. We summarize the cross-sectional $R^2_{OOS,i}$ distribution in Table 2 below.

[Table 2 about here.]

Not surprisingly, RF can challenge the "Oracle" estimator in this nonlinear setting. The NNET is again performing poorly compared to RF. One possible reason is that the regression coefficients are too small in the data generating process, such that our implemented NNET cannot detect the signal. We remark that the global R^2_{OOS} can be slightly misleading when comparing the linear methods to the intercept-only regression. Looking at the security-specific $R^2_{OOS,i}$, we see that the performance of the Average is similar to the performance of the linear models. In contrast, the global R^2_{OOS} indicates that the linear methods outperform the Average by a large margin.

3.5 Sparsity

In the age of big data, it is important to tame the factor zoo and screen for the relevant driving factors. Often, the data generating process is assumed (and empirically observed) to be highly sparse and noisy such that model-selection is not straightforward. Therefore, we next examine the influence of the number of driving factors K^* on the prediction accuracy. For different

choices of K^*, we summarize the cross-sectional $R^2_{OOS,i}$ distribution in Table 3. For comparison, we include the base-case results for $K^* = 3$.

We observe that VASA and LASSO perform best for all choices of $K^* \in \{1, 3, 10\}$. Having in mind the link between dropout regression, RIDGE, and VASA, it is interesting to see that VASA indeed performs similar to RIDGE and LASSO and challenges even the "Oracle" estimator. All methods outperform the Average in every scenario. Note that the performance ranking across methods is very stable in terms of the number of driving covariates. Additionally, we find that the prediction accuracy increases rapidly with a higher number of influential covariates.

[Table 3 about here.]

3.6 Signal-to-Noise Ratio

It is common knowledge that for weak signals and strong noise, the (penalized) linear models struggle to predict stock returns. Consequently, we change the signal to noise parameter from almost no signal (base-case), $\theta_0 = 0.02$, to a "very" strong signal, $\theta_0 = 0.1$. By doing so, we can analyze for which scenarios VASA can improve prediction accuracy and variability and where it has its limitations. The cross-sectional $R^2_{OOS,i}$ distribution is summarized in Table 4.

[Table 4 about here.]

VASA has a very high prediction accuracy and low variability across all signal-to-noise ratios, similar to the penalized linear methods. It performs best for a strong signal, $\theta_0 = 0.1$, although the difference to LASSO is small. Interestingly, the performance of OLS is similar to the performance of the penalized methods. Although P is large, the OLS method profits from the fact that we have $N \times T \gg P$. If this is the case, OLS performs well in a panel regression setting.

Inspecting the performance of the nonlinear methods in Table 4, we find that their performance stays behind the penalized linear methods and VASA. This observation comes at no surprise since we assume a linear data generating process. Nevertheless, when we increase the regression coefficient from $\theta_0 = 0.02$ to $\theta_0 = 0.1$ in the data generating process, we observe that the performance of the NNET starts to improve. For $\theta_0 = 0.1$, it even outperforms RF.

As an intermediate conclusion from the analysis of our simulation exercise, we find that our VASA performs remarkably well and is the most robust and accurate model when the quality of a signal is unknown, which is usually the case in practice. First, it performs well if there is almost no signal at all, as it reduces noise overfitting by aggregating submodel predictions. Second, it performs well if there are strong signals, as the subsampling probabilities take into account the variable importance. Lastly, it performs well for other nonlinear and sparse scenarios, as VASA controls for model-selection mistakes by aggregating submodel predictions.

4 Empirical Analysis

Before we empirically analyze the performance of the different prediction methods, we discuss the data and how we split the sample into training, validation, and out-of-sample period.

4.1 Data

We download monthly stock return data from the Center for Research in Security Prices (CRSP). Our sample period starts in January 1977 and ends in December 2016, totaling 40 years. Also, we obtain the 94 stock-level predictive characteristics used by Gu et al. (2020) and industry dummies corresponding to the first two digits of Standard Industrial Classification (SIC) codes from Dacheng Xiu's webpage.[24] Tables 5–6 lists all the 94 stock-level predictive characteristics and their corresponding main literature. To compute an informative stock-level prediction accuracy measure and not just an aggregated global predictive R^2, we restrict our sample to stocks that have a complete return and stock-level characteristics history for the entire 40 years. In doing so, the number of stocks in our sample reduces to 501. We also obtain the Treasury-bill rate to proxy for the risk-free rate from which we calculate individual excess returns.

Furthermore, we construct eight macroeconomic predictors following the variable definitions detailed in Welch and Goyal (2008), including dividend-price ratio (dp), earnings-price ratio (ep), book-to- market ratio (bm), net equity expansion (ntis), Treasury-bill rate (tbl), term spread (tms), default spread (dfy), and stock variance (svar). The monthly data are available from Amit Goyal's website.[25]

[24] See, http://dachxiu.chicagobooth.edu.
[25] See, http://www.hec.unil.ch/agoyal.

As in Gu et al. (2020), we distinguish between two sets of covariates. The first set consists of stock-level covariates based on the 94 stock characteristics

$$\boldsymbol{z}_{i,t}^{\text{standard}} := \boldsymbol{c}_{i,t} \ , \tag{4.1}$$

where $\boldsymbol{c}_{i,t}$ is the 94×1 vector of characteristics for each stock i at t. The second and the larger set of stock-level covariates includes also the interactions between the 8 macroeconomic state variables $\boldsymbol{x}_t$ and the stock-level characteristics as well as 58 industry dummies $\boldsymbol{d}_{i,t}$

$$\boldsymbol{z}_{i,t}^{\text{large}} := \begin{pmatrix} \boldsymbol{c}_{i,t} \\ \boldsymbol{x}_t \otimes \boldsymbol{c}_{i,t} \\ \boldsymbol{d}_{i,t} \end{pmatrix} . \tag{4.2}$$

Hence, the total number of covariates is $94 \times (1 + 8) + 58 = 904$.

[Table 5 about here.]

[Table 6 about here.]

We divide the 40 years of data into 12 years of training sample (1977 – 1988), eight years of validation sample (1989 – 1996), and the remaining 20 years (1997 – 2016) for out-of-sample testing using the hybrid sample splitting scheme described in Section 2.3.

4.2 Return Prediction

When we estimate a panel model $g_i(\cdot) \equiv g(\cdot)$, we can increase the sample size by the factor N, which reduces the estimation error and leads to more robust results. This effect is only possible because we work with stock characteristics and not with common factors. In contrast to Gu et al. (2020), we predict only stock returns for stocks with no missing returns $N = 501$. Focusing on this universe of stocks has several advantages. First, we can apply a panel-data analysis to historical data, which is consistent with our simulation analysis. Second, we can compute a fair stock-specific prediction accuracy measure, in this case, $R^2_{OOS,i}$, which helps us emphasizing the common pitfalls when relying only on the global R^2_{OOS}. Third, we can avoid any further assumptions about data cleaning, which may introduce further biases.

Table 7 presents the comparison of machine learning techniques in terms of their individual and global out-of-sample predictive R^2, where we distinguish between the standard set of predictive signals $\boldsymbol{z}_{i,t}^{\text{standard}}$ based on the 94 stock-level characteristics, and the large set of 904 predictive signals $\boldsymbol{z}_{i,t}^{\text{large}}$.[26] We make the following observations. Except for OLS in the large-dimensional setting, all models generate a positive global R^2_{OOS}, indicating a substantial outperformance over the naive forecast of zero. Additionally, in most cases, the R^2_{OOS} values are much higher than those reported in the analysis of Gu et al. (2020). Arguably, the significant increase in R^2_{OOS} is due to the selected investment universe, in which we exclude stocks with missing returns and characteristics and keep only those stocks that are available for the whole sample.

[Table 7 about here.]

While it comes at no surprise that the OLS model fails in the large-dimensional case, as the lack of regularization leaves OLS highly susceptible to in-sample overfit, it performs reasonably well for the 94 stock characteristics. Thus, as long as the dimension is not too large for OLS, a simple linear model provides a remarkable aggregate out-of-sample predictive R^2, slightly outperforming RF in terms of R^2_{OOS}. Hence, overall for the 94 stock characteristics, the performance of linear and nonlinear models is comparable, with VASA providing the highest R^2_{OOS}.

When we move to the large-dimensional setting, it becomes clear that we should restrict OLS to a sparse parameterization. For example, forcing the model to include only three covariates like size, value, and momentum, we find that the historical average is a hard benchmark to beat. However, regularizing the linear model via dimension reduction (LASSO, RIDGE, VASA) generates a substantial improvement over the full and sparse OLS models. RF more than doubles the performance in terms of the traditional global R^2_{OOS} (2.53%), leaving all other linear models behind. However, NNET also performs well in the large-dimensional case, almost as good as RF in terms of the global R^2_{OOS}.

The results in Table 7 seem to indicate that for the baseline model with 94 stock characteristics,

[26] We find that the performance of VASA is not sensitive to the number of submodels used. After some averaging over submodels, let us say five, there is almost no benefit of estimating additional submodels. It seems that the prediction accuracy of any panel-data submodel with optimal subsampling size (κ somewhere between 8 and 16) is very similar. This finding is good news, as B needs not to be very large, already $B = 10$ would be enough.

VASA is our method of choice. At the same time, with a large number of predictive signals, in this case, 904, the nonlinear RF should be preferred to the linear methods. However, we argue that the global R^2_{OOS} is a misleading performance measure that can lead to wrong conclusions. The reason is the following. The global R^2_{OOS} summarizes the mean cross-sectional performance but completely neglects the stock-specific risk of each prediction. In particular, a model can have a high global R^2_{OOS}, and, on average, it predicts stock returns with reasonable accuracy. Still, for some stocks, the predictions may be extremely far from the truth. Therefore, we propose to take into account the distribution of the stock-specific $R^2_{OOS,i}$ to investigate not only the mean performance but also the prediction risk for each model.

[Figure 1 about here.]

In Figure 1, we provide the boxplots of the stock-specific $R^2_{OOS,i}$'s for the different methods and for both the standard and the large-dimensional covariate vectors $\boldsymbol{z}_{i,t}$. The dispersion of the individual stock return prediction accuracy varies substantially across the different methods. While the aggregate prediction accuracy, measured by AV and MED, is the driving factor for the global R^2_{OOS}, the high variability across stocks plays an important role when applied in practice. When using the 94 stock characteristics, we find a similar level and variability of individual $R^2_{OOS,i}$. VASA seems to be the best performing method with the highest level (AV, MED, and R^2_{OOS}) and relatively low variability (SD) as well as low risk of negative outliers (P10).

When we use the 904 predictive signals, the comparison of the global R^2_{OOS} in Table 7 suggests that we use RF with a large set of predictors. However, Figure 1 clearly illustrates that, compared to the other models, RF also exposes us to a substantial risk that the prediction based on RF will fail for a specific stock and that we get an 'unrealistically' high or low return prediction.[27] Consequently, taking into account the global R^2_{OOS} fails to address some essential aspects for return prediction, and we should also consider the level and variability of the individual $R^2_{OOS,i}$ in our empirical analysis. Taking these considerations into account, it seems that also for the high dimensional case, VASA provides a reasonable trade-off between prediction accuracy and prediction variability.

[27]The prediction risk inherent in the RF method is reflected in the high variability (SD = 5.24%) and the occurrence of extreme negative outliers (P10 = -4.53%). Note that in the large-dimensional setting the ultra-sparse models (Average, OLS-3) are almost impossible to beat in terms of variability.

In Figure 1, we observe that the results for NNET in the high-dimensional case seem to be promising. AV and MED are higher than those from RF, and the variability is substantially lower. Hence, to sum up, based on the analysis of both out-of-sample R^2 measures (R^2_{OOS} and $R^2_{OOS,i}$) it is difficult to say if the inclusion of interaction terms and industry dummies can improve prediction accuracy. On the one hand, 'smart' (penalized) linear and nonlinear models can increase the average performance. On the other hand, they have a higher risk for individual stocks. From the above analysis, it is still not clear how this trade-off between prediction accuracy and prediction variability will impact portfolio performance. Hence, in the next section, we investigate this question and analyze how accuracy and variability impact portfolio returns.

4.3 Performance Analysis

So far, our assessment of forecast performance has been entirely statistical, relying on comparisons of individual and global predictive R^2. In this section, we form long-short portfolios based on different machine learning forecasts and analyze their performance. At the end of each month, we calculate one-month-ahead out-of-sample return predictions for each method. We then sort stocks into deciles based on each model's forecasts. We reconstitute portfolios each month using equal weights. Finally, we construct a zero-net-investment portfolio that buys the highest expected return stocks (top decile) and sells the lowest (bottom decile).[28] Whenever the top (bottom) decile includes returns with negative (positive) returns, we will replace these returns with a long (short) position in the money market account.

Since our statistical objective functions minimize equally-weighted forecast errors, we construct equally-weighted long-short portfolios and analyze their performance. As a robustness check, we also report the value-weighted long-short portfolio performance. Value-weighted portfolios are less sensitive to trading costs and small-cap biases. As an additional robustness test, we consider the efficient sorting approach of Ledoit et al. (2019) to generate a more robust zero-net-investment portfolio taking into account the information from the covariance matrix of the investment universe.

[Table 8 about here.]

[28] Note that 'Average' gives the same prediction for each share, and thus, no sorting can be applied for this method. Nevertheless, we interpret the 'Average' method as the $1/N$ or equally-weighted portfolio and include it as a further benchmark even though it has no zero-net-investment.

Table 8 reports the out-of-sample results for the equal-weighted long-short portfolios when we use $\boldsymbol{z}_{i,t}^{\text{standard}}$ and $\boldsymbol{z}_{i,t}^{\text{large}}$. For $\boldsymbol{z}_{i,t}^{\text{standard}}$ as predictor, all models consistently outperform the Average (often by a wide margin). Additionally, VASA consistently and markedly outperforms all other models in terms of AV, leading to the largest final portfolio value. VASA also exhibits the most favorable portfolio kurtosis. While VASA has a significantly lower SD than LASSO, RIDGE, RF, and NNET, the lowest SDs are delivered by the Average and OLS. Consequently, VASA generates the second-largest Sharpe ratio (SR=1.224). Surprisingly, the highest Sharpe ratio (SR=1.239), but only by a small margin, is not generated by a more sophisticated model like RF or NNET, but by OLS. The Sharpe ratios of RF and NNET stay below 1. However, their advantages over simple methods like OLS become more obvious when we move from $\boldsymbol{z}_{i,t}^{\text{standard}}$ to $\boldsymbol{z}_{i,t}^{\text{large}}$ and increase the number of predictors to 904.

When we use $\boldsymbol{z}_{i,t}^{\text{large}}$ as a predictor, we indeed find that RF and NNET provide a better Sharpe ratio than the simple linear models OLS and OLS-3. As OLS makes no dimension reduction, it comes as no surprise that it is better to focus only on the 94 stock-level characteristics instead of the 904 covariates. Whereas a simple regression performs surprisingly well for $\boldsymbol{z}_{i,t}^{\text{standard}}$ it delivers a worse but still plausible portfolio for $\boldsymbol{z}_{i,t}^{\text{large}}$, similar than Average or RF. Note that even though simple OLS delivers reasonable equally-weighted long-short portfolios for the investigated investment universe, the predicted portfolio returns are unrealistically high due to the highly negative R^2_{OOS} values; see Table 7. For the 904 covariates, VASA delivers the highest Sharpe ratio (SR=1.480) by a large margin over the second-largest Sharpe ratio generated by NNET (SR=1.137). Again, VASA generates the highest average return (AV=34.271%), compared to RIDGE (AV=26.191%) and NNET (AV=21.424%).

[Figure 2 about here.]

In Figure 2, we compare the evolution of the different portfolios when we use $\boldsymbol{z}_{i,t}^{\text{standard}}$ to form our predictions. We find that VASA is the best performing strategy, closely followed by LASSO, NNET, and RIDGE. Surprisingly, OLS beats the more sophisticated RF. However, all strategies beat the Average, i.e., the $1/N$-portfolio is the worst performing portfolio. Note, however, that the Average is a long-only portfolio. Therefore, in the lower panel of Figure 2, we separately plot the long and short lag of the different portfolio strategies. While all strategies perform equally well on the long lag, significant differences emerge when we look at the short lag.

While VASA, LASSO, and NNET perform well, RF fails to construct a successful short strategy.

[Figure 3 about here.]

Figure 3 provides the evolution of portfolio values, when using $\boldsymbol{z}_{i,t}^{\text{large}}$ as predictor. Clearly, in terms of the cumulative return (in log-scale), VASA significantly outperforms all other methods. RIDGE turns out to be the second-best method, while LASSO and NNET end up at approximately the same value at the end of the sample period. In the lower panel of Figure 2, we observe that the long lag of each method outperforms the $1/N$-strategy as well. The long lag of LASSO, RIDGE, VASA, and NNET generate similar or almost identical final values. However, while LASSO, RIDGE, and in particular, NNET struggle to generate a successful short-lag strategy, the outperformance of VASA's long-short portfolio seems to profit not only on the well-performing long lag but also on the successful construction of the short lag.

From the above analysis, we see that linear regression models using dimension reduction via penalization (RIDGE, LASSO) or subsampling (VASA), perform well in both scenarios. So does NNET. However, especially VASA can benefit from the additional macroeconomic and industry information, which helps VASA to improve the performance contribution from the short lag. These findings provide a further indication that the global R^2_{OOS} can be a misleading measure. RF has by far the highest global R^2_{OOS} value for $\boldsymbol{z}_{i,t}^{\text{large}}$, but the performance suffers when we use these predictions to compute the long and short positions of a portfolio. We argue that this is due to the high risk induced by RF's prediction performance. As discussed in the previous section, RF has the highest individual $R^2_{OOS,i}$ volatility, at least for the 501 stocks and $\boldsymbol{z}_{i,t}^{\text{large}}$. Thus, even though RF predicts stock returns the best (highest R^2_{OOS}) on average, it performs badly for some stocks, giving RF the highest SD and lowest P10 values. Thus, when creating long and short positions for the RF portfolio, we run a high risk of misclassifying stocks into the wrong decile.

The question is now, why do RF and (partially) NNET fail to predict stock and long-short portfolio returns compared to more straightforward linear (dimension reduction) models?[29] Our observation seems puzzling, as the empirical analysis of Gu et al. (2020) shows the power of the

[29]Whereas the results for RF are very robust regarding hyperparameter tuning, we do not claim that there are no better and smartly tuned NNETs that can outperform the other machine learning methods. Our main point is that a standard NNET, similar as in Gu et al. (2020), cannot improve stock and long-short portfolio prediction performance for the investigated investment universe.

random forest and neural nets to predict stock and portfolio returns. However, we might get some indication of potential causes of our findings when we take a closer look at what exactly drives their outperformance over linear models. Almost all the Sharpe ratio gains come from the short position, while the performance of the long position across the methods is similar. In their analysis, only random forest and neural nets can generate gains, and thus positive Sharpe ratio, from going short. This observation means that more sophisticated machine learning algorithms seem to be significantly better in evaluating which stocks perform poorly in the next month for *all* available shares in the market.

However, in our setting, where we analyze not the entire market but those stocks that have a complete return and characteristics history, we find the opposite; see Figure 3. Whereas the performance of the long position is still similar across the methods, RF and NNET are not capable anymore of generating gains from the short position. Only VASA has a positive Sharpe ratio also for the short position. Arguably, the worse performance for RF and NNET in the short position, compared to Gu et al. (2020, Table 7), may stem from the choice of our investment universe. We consider only stocks that have a full data history. Thus, they have survived the last 40 years without any bankruptcy, merger and acquisition, renaming, and similar events. Arguably, RF and NNET struggle to find the bottom decile of these stocks to go short as they tend to be more stable and secure stocks. Thus, the signal is too weak to generate a decent short portfolio.

If we compare our level of SR with the results of Gu et al. (2020, Table 7), we get, as discussed before, a worse performance for RF and NNET but better for penalized linear models where VASA (1.48) even outperforms their best NNET (1.35). Hence, we are left puzzled by the fact that the results of Gu et al. (2020), which favor RF and NNET, appear to be mainly driven by the stocks for which no entire history is available. If RF and NNET were stable algorithms, we would have expected that they also outperform simpler linear models for our data set. We leave a more in-depth analysis of this observation for future research. One of our main points for this paper is that we want to raise a warning flag on using a global predictive R^2 for selecting different methods. As Table 7 indicates, the R^2_{OOS} of NNET is more than twice as large as the R^2_{OOS} of VASA. However, the variance of the $R^2_{OOS,i}$ of NNET is also more than three times larger than the one of VASA. This variation in $R^2_{OOS,i}$ has such a negative impact on NNET's

portfolio performance that it fails to outperform VASA.

4.4 Robustness Checks

To further robustify our results, we do the following exercise. First, we use different percentiles to form our portfolio sorts. Second, we analyze portfolio performance when we use different weighting strategies. To analyze the impact on portfolio performance when using different percentiles, we focus on the portfolios based on $\boldsymbol{z}_{i,t}^{\text{large}}$, and we use equal-weights within the percentiles. So far, we have used the 10th percentile to construct the short and the long portfolios. In what follows, we use the 5th as well as the 30th percentiles. Figure 4 provides an overview of the results. It turns out that the ranking of the different strategies does not change significantly. Again, VASA outperforms all models. Hence, our results are robust for different choices of percentiles.

[Figure 4 about here.]

All the methods analyzed above minimize equally-weighted forecast errors, which is the most appropriate statistical objective function when we aim at constructing equal-weighted portfolios. Nevertheless, it is instructive to analyze whether our results are robust to different weighting schemes. As a first obvious choice for an alternative weighting scheme, we change from equal-weighted to value-weighted portfolio sorts. Table 9 summarizes the results.

[Table 9 about here.]

When using $\boldsymbol{z}_{i,t}^{\text{standard}}$ as a predictor, the largest Sharpe ratios are generated by RIDGE (SD=1.013) and VASA (SD=1.004), with a small advantage for RIDGE. However, when moving to $\boldsymbol{z}_{i,t}^{\text{large}}$, the highest Sharpe Ratio (SR=1.339), again by a wide margin, is provided by VASA. Hence, VASA profits the most from enlarging the set of predictors. At the same time, the Sharpe Ratio for RF deteriorates substantially. Figure 5 plots the corresponding evolution of the cumulative log-returns of the different strategies. Clearly, for $\boldsymbol{z}_{i,t}^{\text{large}}$, VASA turns out to be the superior strategy.

[Figure 5 about here.]

As an additional weighting alternative, we use efficient sorting approach introduced by Ledoit et al. (2019). The basic idea behind this method is to exploit the information from the covariance matrix. In particular, our goal is to minimize portfolio variance under the constraint that the resulting portfolio has the same expected return as the strategy based on the long-short portfolio sorts. Hence, we need to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \boldsymbol{w}' \widehat{\Sigma}_{t+1} \boldsymbol{w} \tag{4.3}$$

subject to

$$\mathbb{E}_t[\boldsymbol{r}_{t+1}]' \boldsymbol{w} = \mathbb{E}_t[\boldsymbol{r}_{t+1}]' \boldsymbol{w}_{t+1}^{\text{EW}}$$
$$\sum_{w_i<0} |w_i| = \sum_{w_i>0} |w_i| = 1 \ .$$

By $\widehat{\Sigma}_{t+1}$ we denote a feasible estimator of the (conditional) covariance matrix of the stock returns and $\boldsymbol{w}_{t+1}^{\text{EW}}$ is the weight vector of the equally-weighted portfolio based on sorting, as we have used in Section 4.2 for the different forecasting methods. In our empirical analysis, we use a static estimator of the large-dimensional covariance matrix $\widehat{\Sigma}$ based on the analytical nonlinear shrinkage of Ledoit and Wolf (2020).[30]

[Figure 6 about here.]

Figure 6 confirms previous findings. For both predictors, VASA generates the highest portfolio value at the end of our sample period. Since efficient sorting aims at minimizing the variance for a given target return, the Sharpe ratio of the resulting portfolios tends to be higher than the ones based on portfolio sorts. This goal is indeed achieved and confirmed by the data. As we see from Table 10, we can increase the Sharpe ratio of VASA to 1.908, compared to the Sharpe ratio of 1.480 for the equal-weighted strategy.

[Table 10 about here.]

[30] In unreported results, we find that dynamic estimators such as the DCC-NL of Engle et al. (2019) and AFM-DCC-NL of De Nard et al. (2020) can further decrease the out-of-sample standard deviation.

4.5 VASA's Factor Choice

As in Gu et al. (2020), we are interested in which variables are selected by the different methods. Since all methods except VASA have been discussed in their paper, and since our results do not differ, we exclusively focus on VASA and ask which submodels are chosen. For that purpose, we look at how VASA chooses both stock-level characteristics and interaction terms. To this end, we order all the covariates based on their total frequency of all VASA submodels over time, with the most frequent covariates on top and least frequent on the bottom.

[Figure 7 about here.]

Figure 7 plots the time variation in the ranks of the different covariates. The top-ranked characteristics, there is not much fluctuation across time. The most critical characteristics selected by VASA are stable over time. Momentum turns out to be the most influential factor, especially when enriched with an interaction term. We recall that, for VASA, the choice of the submodels are based on the in-sample R^2's, which we use to calculate the subsampling probabilities. If we would subsample with equal probabilities, then covariates in Figure 7 would have approximately the same color. Hence, by using the in-sample R^2, VASA generates a ranking that looks very similar to the ranking obtained by Gu et al. (2020, Figure A.5) using their neural net. We also remind that VASA builds submodels with around 15 to 20 covariates. As we see from Figure 7, from the 20 most important covariates, only three of them are pure stock-level characteristics. All other covariates include macroeconomic interaction terms. Moreover, from the 100 most important covariates, we have only eleven pure stock-level characteristics. Hence, most of the submodels in VASA are driven by characteristics with interaction terms.

5 Conclusion

We perform a comparative analysis of machine learning algorithms for predicting equity returns using large-dimensional factor models. We demonstrate that more sophisticated algorithms like random forest and neural networks do not necessarily beat simpler linear models. Although neural networks are hugely popular in a wide range of research disciplines, it is well-known that they may generate unstable predictions. By switching from a global comparison of prediction

accuracy (R^2_{OOS}) to a security-specific prediction accuracy measure ($R^2_{OOS,i}$), we can shed some additional light on this issue. In particular, we find that the variability of $R^2_{OOS,i}$'s can be substantial for random forests and neural nets, while for our newly proposed method based on variable subsampling aggregation, VASA, has substantial lower prediction risk. We conjecture that high variability in $R^2_{OOS,i}$'s is detrimental for long-short portfolios sorted according to predicted returns, due to the higher risk of misclassifying the stocks in the wrong deciles.

In an empirical exercise, we find our conjecture confirmed. The low variability of VASA's security-specific $R^2_{OOS,i}$'s in the simulation carries over to the historical analysis. VASA outperforms all other methods in terms of the Sharpe ratio. Even the nonlinear methods like random forests and neural nets cannot beat our linear VASA. To further robustify our results, we perform a set of additional checks, but our conclusions did not change. Hence, we show that a large global R^2_{OOS} may still be no guarantee for outperformance. Complex nonlinear models may turn out to be unstable, mainly when applied to situations in which the number of variables is large and multicollinearity may be present.

We track down the source of VASA's predictive advantage to maintain the simple, intuitive linear structure by estimating multiple linear submodels that reduce the curse of dimensionality and to aggregate their predictions to control for model-selection mistakes and nonlinearity. We find that already a naive implementation of VASA can significantly improve prediction accuracy and risk. Additionally, we suggest computing subsampling probabilities based on variable importance measures and submodel predictions aggregation weights via in-sample fit measures. This procedure gives even better out-of-sample predictions by smartly averaging over more realistic subsampled factor models.

In this paper, we have introduced VASA as a simple method for prediction and we have shown its advantages. However, VASA is not restricted to linear submodels, and future research should focus on more complex nonlinear base algorithms and aggregation functions.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abarbanell, J. and Bushee, B. (1998). Abnormal returns to a fundamental analysis strategy. *The Accounting Review*, 73(1):19–45.

Ahmad, M. W., Mourshed, M., and Rezgui, Y. (2017). Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption. *Energy and Buildings*, 147:77 – 89.

Ali, A., Hwang, L., and Trombley, M. (2003). Arbitrage risk and the book-to-market anomaly. *Journal of Financial Economics*, 69(2):355–373.

Almeida, H. and Campello, M. (2007). Financial constraints, asset tangibility, and corporate investment. *The Review of Financial Studies*, 20(5):1429–1460.

Amihud, Y. (2002). Illiquidity and stock returns: cross-section and time-series effects. *Journal of Financial Markets*, 5(1):31–56.

Amihud, Y. and Mendelson, H. (1989). The effects of beta, bid-ask spread, residual risk, and size on stock returns. *The Journal of Finance*, 44(2):479–486.

Anderson, C. and Garcia-Feijóo, L. (2006). Empirical evidence on capital investment, growth options, and security returns. *The Journal of Finance*, 61(1):171–194.

Ang, A., Hodrick, R. J., Xing, Y., and Zhang, X. (2006). The cross-section of volatility and expected returns. *Journal of Finance*, 61(1):259–299.

Asness, C., Porter, B., and Stevens, R. (2000). Predicting stock returns using industry-relative firm characteristics. Working paper.

Balakrishnan, K., Bartov, E., and Faurel, L. (2010). Post loss/profit announcement drift. *Journal of Accounting and Economics*, 50(1):20–41.

Bali, T. G., Cakici, N., and Whitelaw, R. F. (2011). Maxing out: Stocks as lotteries and the cross-section of expected returns. *Journal of Financial Economics*, 99(2):427–446.

Bandyopadhyay, S. P., Huang, A. G., and Wirjanto, T. S. (2010). The accrual volatility anomaly. Working paper, School of Accounting and Finance, University of Waterloo.

Banfield, R. E., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. (2006). A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):173–180.

Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics*, 9(1):3–18.

Barbee, W., Mukherji, S., and Raines, G. (1996). Do sales-price and debt-equity explain stock returns better than book-market and firm size? *Financial Analysts Journal*, 52(2):56–60.

Barth, M., Elliott, J., and Finn, M. (1999). Market rewards associated with patterns of increasing earnings. *Journal of Accounting Research*, 37(2):387–413.

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *Journal of Finance*, 32(3):663–682.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139.

Belo, F., Lin, X., and Bazdresch, S. (2014). Labor hiring, investment, and stock return predictability in the cross section. *Journal of Political Economy*, 122(1):129–177.

Bhandari, L. C. (1988). Debt/equity ratio and expected common stock returns: Empirical evidence. *Journal of Finance*, 43(2):507–528.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brown, D. and Rowe, B. (2007). The productivity premium in equity returns. Working paper.

Bühlmann, P., Hothorn, T., et al. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22(4):477–505.

Campbell, J. Y. and Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.

Chandrashekar, S. and Rao, R. K. (2009). The productivity of corporate cash holdings and the cross-section of expected stock returns. *McCombs Research Paper Series No. FIN-03-09*.

Chen, L., Pelger, M., and Zhu, J. (2019). Deep learning in asset pricing. *Available at SSRN 3350138*.

Chordia, T., Subrahmanyam, A., and Anshuman, V. R. (2001). Trading activity and expected stock returns. *Journal of Financial Economics*, 59(1):3–32.

Cochrane, J. H. (2011). Presidential address: Discount rates. *The Journal of Finance*, 66(4):1047–1108.

Cooper, M. J., Gulen, H., and Schill, M. J. (2008). Asset growth and the cross-section of stock returns. *Journal of Finance*, 63(4):1609–1651.

Coqueret, G. and Guida, T. (2018). Stock returns and the cross-section of characteristics: A tree-based approach. *Available at SSRN 3169773*.

Datar, V. T., Naik, N. Y., and Radcliffe, R. (1998). Liquidity and stock returns: An alternative test. *Journal of Financial Markets*, 1(2):203–219.

De Nard, G., Ledoit, O., and Wolf, M. (2020). Factor models for portfolio selection in large dimensions: The good, the better and the ugly. *Journal of Financial Econometrics*, forthcoming.

Desai, H., Rajgopal, S., and Venkatachalam, M. (2004). Value-glamour and accruals mispricing: One anomaly or two? *The Accounting Review*, 79(2):355–385.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Dietterich, T. G. (2000a). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157.

Eberhart, A. C., Maxwell, W. F., and Siddique, A. R. (2004). An examination of long-term abnormal stock returns and operating performance following R&D increases. *Journal of Finance*, 59(2):623–650.

Eisfeldt, A. and Papanikolaou, D. (2013). Organization capital and the cross-section of expected returns. *Journal of Accounting Research*, 68(4):1365–1406.

Engle, R. F., Ledoit, O., and Wolf, M. (2019). Large dynamic covariance matrices. *Journal of Business & Economic Statistics*, 37:363–375. doi: 0.1080/07350015.2017.1345683.

Fairfield, P., Whisenant, S., and Yohn, L. (2003). Accrued earnings and growth: Implications for future profitability and market mispricing. *The Accounting Review*, 78(1):353–371.

Fama, E. and MacBeth, J. (1973). Risk, return, and equilibrium: Empirical tests. *The Journal of Political Economy*, 81(3):607–636.

Fama, E. F. and French, K. R. (2015). A five factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Francis, J., LaFond, R., Olsson, P., and Schipper, K. (2004). Costs of equity and earnings attributes. *The Accounting Review*, 79(4):967–1010.

Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *Review of Financial Studies*, (forthcoming).

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Gettleman, E. and Marks, J. M. (2006). Acceleration strategies. *SSRN Working Paper Series*.

Green, J., Hand, J., and Zhang, F. (2017). The characteristics that provide independent information about average US monthly stock returns. *The Review of Financial Studies*, 30:4389–4436.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, forthcoming.

Guo, R., Lev, B., and Shi, C. (2006). Explaining the short- and long-term ipo anomalies in the us by r&d. *Journal of Business Finance and Accounting*, 33.

Hafzalla, N., Lundholm, R., and Matthew Van Winkle, E. (2011). Percent accruals. *Accounting Review*, 86(1):209–236.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951.

Harvey, C. and Ferson, W. (1999). Conditioning variables and the cross-section of stock returns. *Journal of Finance*, 54:1325–1360.

Harvey, C. R., Liu, Y., and Zhu, H. (2016). ... and the cross-section of expected returns. *The Review of Financial Studies*, 29(1):5–68.

Hastie, T., Tibshirani, R., and Freedman, J. (2009). *The Elements of Statistical Learning*. Springer, New York, 2009 edition. Availabe freely online.

Hastie, T., Tibshirani, R., and Tibshirani, R. J. (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hertz, J., Krogh, A., Palmer, R. G., and Horner, H. (1991). Introduction to the theory of neural computation. *Physics Today*, 44:70.

Holthausen, R. and Larcker, D. (1992). The prediction of stock returns using financial statement information. *Journal of Accounting and Economics*, 15:373–411.

Hong, H. and Kacperczyk, M. (2009). The price of sin: The effects of social norms on markets. *Journal of Financial Economics*, 93:15–36.

Hou, K. and Moskowitz, T. (2005). Market frictions, price delay, and the cross-section of expected returns. *The Review of Financial Studies*, 18(3):981–1020.

Hou, K. and Robinson, D. (2006). Industry concentration and average stock returns. *The Journal of Finance*, 61(4):1927–1956.

Hou, K., Xue, C., and Zhang, L. (2015). Digesting anomalies: An investment approach. *Review of Financial Studies*, 28(3):650–705.

Hou, K., Xue, C., and Zhang, L. (2018). Replicating anomalies. *The Review of Financial Studies*, page forthcoming.

Huang, A. G. (2009). The cross section of cashflow volatility and expected stock returns. *Journal of Empirical Finance*, 16(3):409–429.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jacobsen, B., Jiang, F., and Zhang, H. (2019). Ensemble machine learning and stock return predictability. In *The 4th International Workshop on Financial Markets and Nonlinear Dynamics (Paris*.

Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48(1):65–91.

Jiang, G., Lee, C., and Zhang, Y. (2005). Information uncertainty and expected returns. *Review of Accounting Studies*, 10:185–221.

Kama, I. (2009). On the market reaction to revenue and earnings surprises. *Journal of Banking and Finance*, 36.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kishore, R., Brandt, M., Santa-Clara, P., and Venkatachalam, M. (2008). Earnings announcements are full of surprises. Working paper.

Lakonishok, J., Shleifer, A., and Vishny, R. W. (1994). Contrarian investment, extrapolation, and risk. *Journal of Finance*, 49(5):1541–1578.

Ledoit, O. and Wolf, M. (2020). Analytical nonlinear shrinkage of large-dimensional covariance matrices". *Annals of Statistics*, forthcoming.

Ledoit, O., Wolf, M., and Zhao, Z. (2019). Efficient sorting: A more powerful test for cross-sectional anomalies. *Journal of Financial Econometrics*, forthcoming.

Lerman, A., Livnat, J., and Mendenhall, R. R. (2008). The high-volume return premium and post-earnings announcement drift. *Available at SSRN 1122463*.

Lev, B. and Nissim, D. (2004). Taxable income, future earnings, and equity values. *The Accounting Review*, 79(4):1039–1074.

Litzenberger, R. and Ramaswamy, K. (1982). The effects of dividends on common stock prices tax effects or information effects? *Journal of Finance*, 37(2):429–443.

Liu, M., Wang, M., Wang, J., and Li, D. (2013). Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and chinese vinegar. *Sensors and Actuators B: Chemical*, 177:970 – 980.

Liu, W. (2006). A liquidity-augmented capital asset pricing model. *Journal of Financial Economics*, 82(3):631–671.

Michaely, R., Thaler, R., and Womack, K. (1995). Price reactions to dividend initiations and omissions: Overreaction or drift? *Journal of Finance*, 50(2):573–608.

Mohanram, P. (2005). Separating winners from losers among lowbook-to-market stocks using financial statement analysis. *Review of Accounting Studies*, 10:133–170.

Moritz, B. and Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. *Available at SSRN 2740751*.

Moskowitz, T. and Grinblatt, M. (1999). Do industries explain momentum? *The Journal of Finance*, 54(4):1249–1290.

Moskowitz, T. and Grinblatt, M. (2010). A better three-factor model that explains more anomalies. *The Journal of Finance*, 65(2):563–594.

Novy-Marx, R. (2013). The other side of value: Good growth and the gross profitability premium. *Journal of Financial Economics*, 108(1):1–28.

Ou, J. and Penman, S. (1989). Financial statement analysis and the prediction of stock returns. *Journal of Accounting and Economics*, 11(4):295–329.

Palazzo, B. (2012). Cash holdings, risk, and expected returns. *Journal of Financial Economics*, 104(1):162–185.

Piotroski, J. D. (2000). Value investing: The use of historical financial statement information to separate winners from losers. *Journal of Accounting Research*, pages 1–41.

Pontiff, J. and Woodgate, A. (2008). Share issuance and cross-sectional returns. *Journal of Finance*, 63(2):921–945.

Rasekhschaffe, K. C. and Jones, R. C. (2019). Machine learning for stock selection. *Financial Analysts Journal*, 75(3):70–88.

Richardson, S. A., Sloan, R. G., Soliman, M. T., and Tuna, I. (2005). Accrual reliability, earnings persistence and stock prices. *Journal of Accounting and Economics*, 39(3):437–485.

Ripley, B. D. (1993). Statistical aspects of neural networks. *Networks and chaos—statistical and probabilistic aspects*, 50:40–123.

Ripley, B. D. and Hjort, N. (1996). *Pattern recognition and neural networks*. Cambridge university press.

Rosenberg, B. (1974). Extra-market components of covariance in security returns. *Journal of Financial and Quantitative Analysis*, 9:263–274.

Rosenberg, B., Reid, K., and Lanstein, R. (1985). Persuasive evidence of market inefficiency. *Journal of Portfolio Management*, 11(3):9–16.

Rossi, A. G. (2018). Predicting stock market returns with machine learning. Technical report, Working paper.

Sloan, R. (1996). Do stock prices fully reflect information in accruals and cash flows about future earnings? (Digest summary). *Accounting Review*, 71(3):289–315.

Soliman, M. T. (2008). The use of dupont analysis by market participants. *Accounting Review*, 83(3):823–853.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014a). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014b). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tang, C., Garreau, D., and von Luxburg, U. (2018). When do random forests fail? In *Advances in Neural Information Processing Systems*, pages 2983–2993.

Thomas, J. and Zhang, F. X. (2011). Tax expense momentum. *Journal of Accounting Research*, 49(3):791–821.

Thomas, J. K. and Zhang, H. (2002). Inventory changes and future returns. *Review of Accounting Studies*, 7(2-3):163–187.

Titman, S., Wei, K. J., and Xie, F. (2004). Capital investments and stock returns. *Journal of Financial and Quantitative Analysis*, 39(04):677–700.

Tuzel, S. (2010). Corporate real estate holdings and the cross-section of stock returns. *The Review of Financial Studies*, 23(6):2268–2302.

Valta, P. (2016). Strategic default, debt structure, and stock returns. *Journal of Financial and Quantitative Analysis*, 51(1):1–33.

Van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1).

Welch, I. and Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, 21:1455–1508.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17.

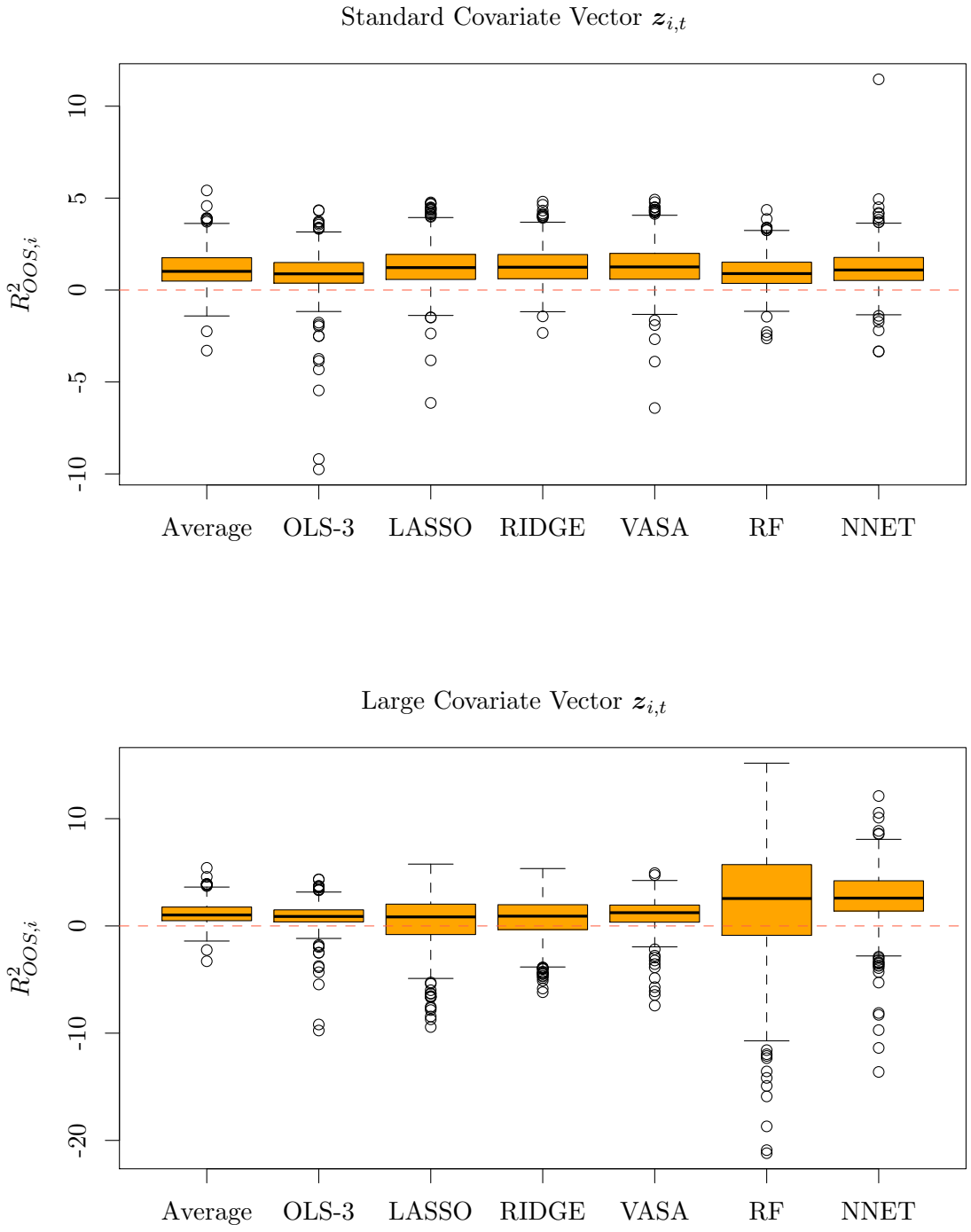

Figure 1: Boxplots of the 501 out-of-sample $R^2_{OOS,i}$ for various methods based on the standard and large-dimensional covariate vector $\boldsymbol{z}_{i,t}$.

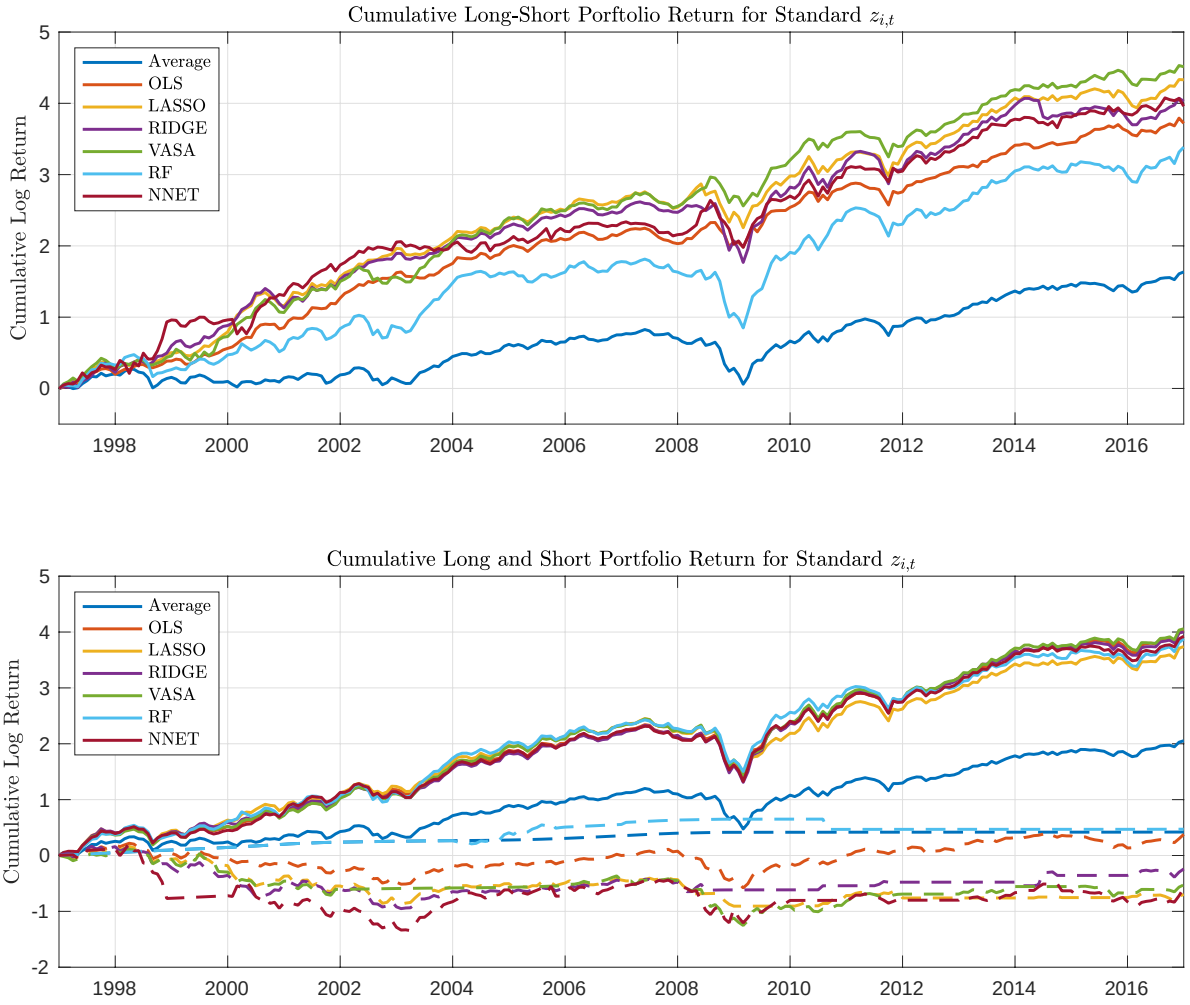

Figure 2: Cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts based on $\boldsymbol{z}_{i,t}^{\text{standard}}$. The solid and dash lines represent long (top decile) and short (bottom decile) positions, respectively. All portfolios are equal-weighted.

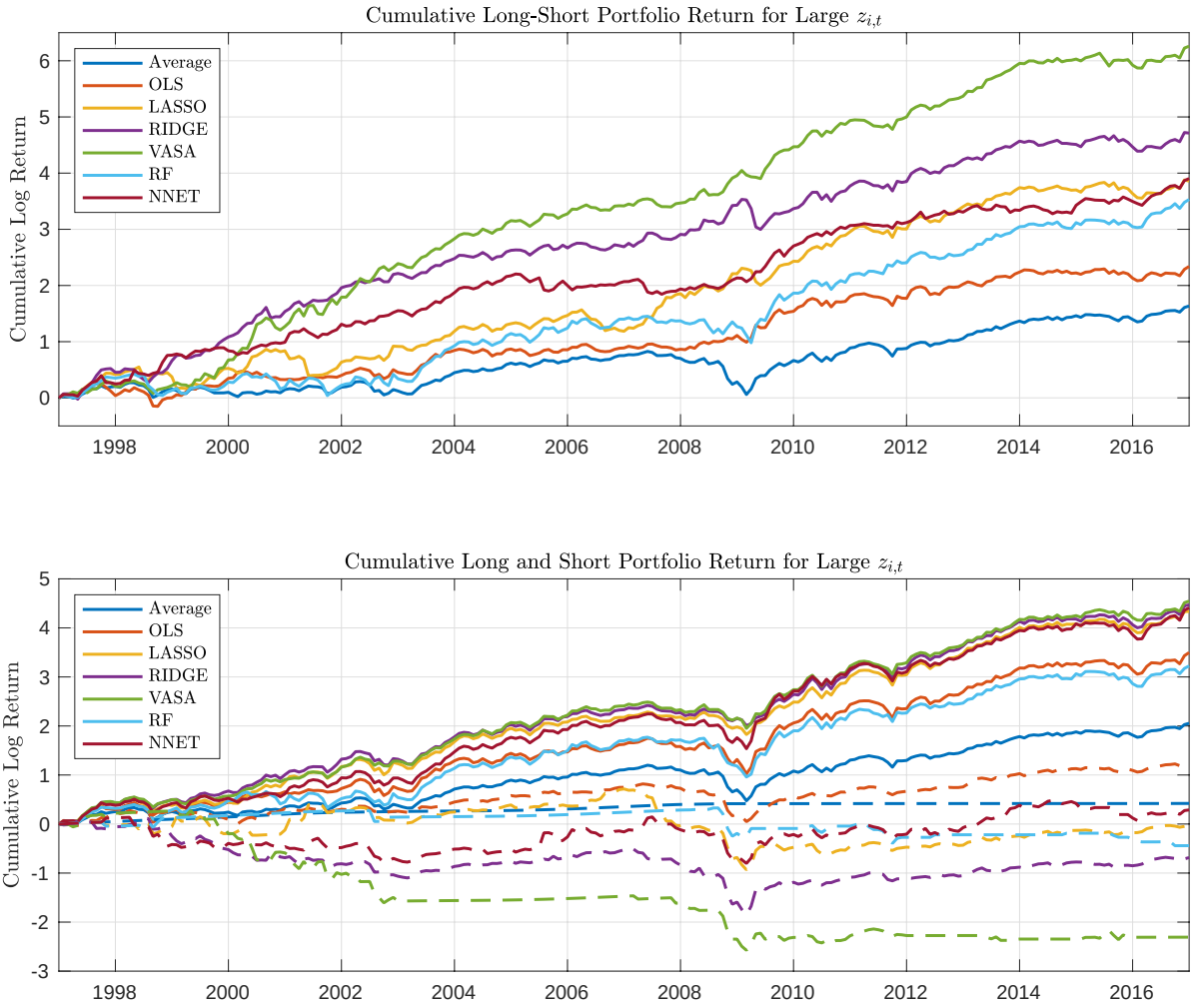

Figure 3: Cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts based on $\boldsymbol{z}_{i,t}^{\text{large}}$. The solid and dash lines represent long (top decile) and short (bottom decile) positions, respectively. All portfolios are equal-weighted.

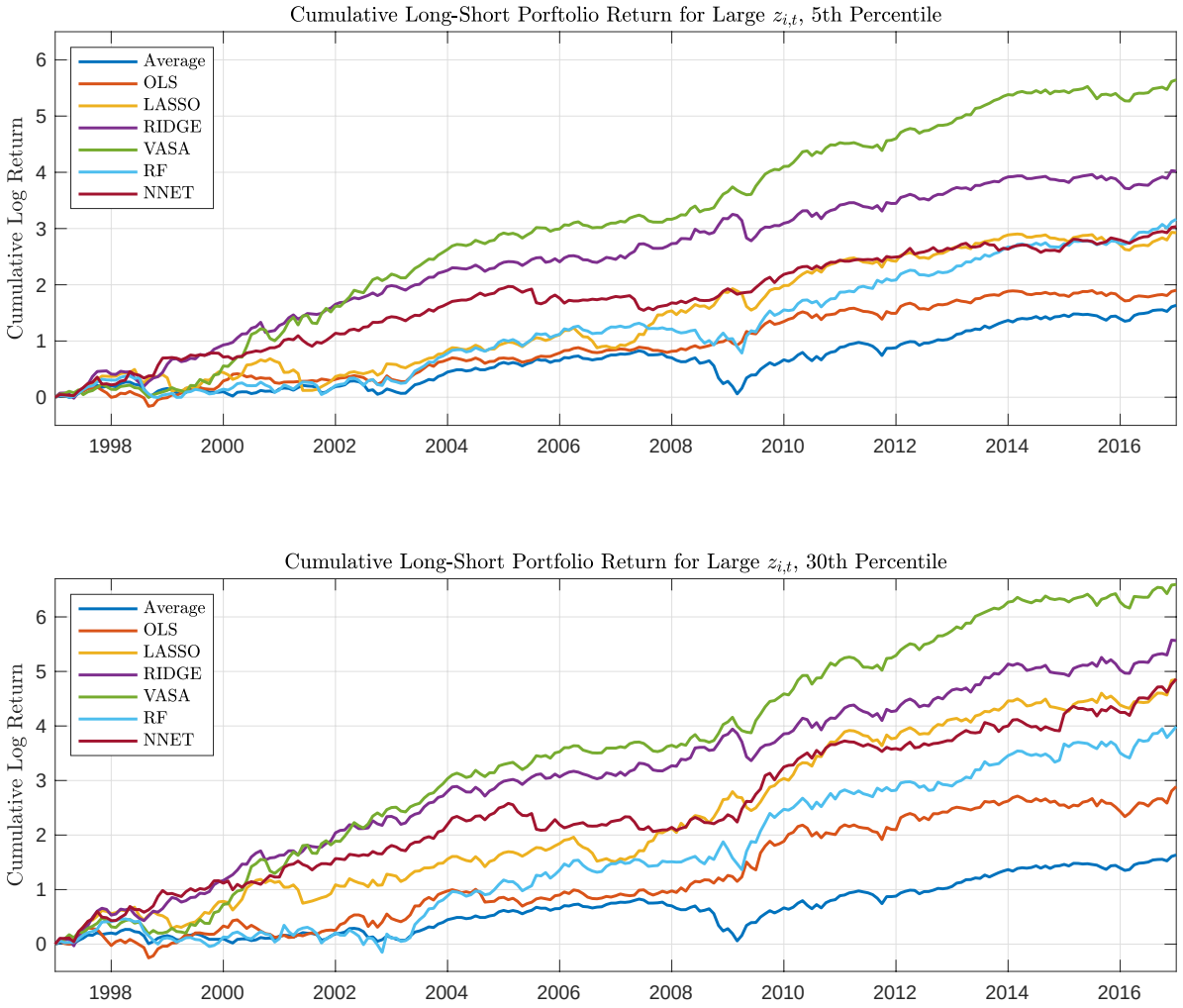

Figure 4: Cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts based on $\boldsymbol{z}_{i,t}^{\text{large}}$ and for different choices of percentiles. All portfolios are equal-weighted.

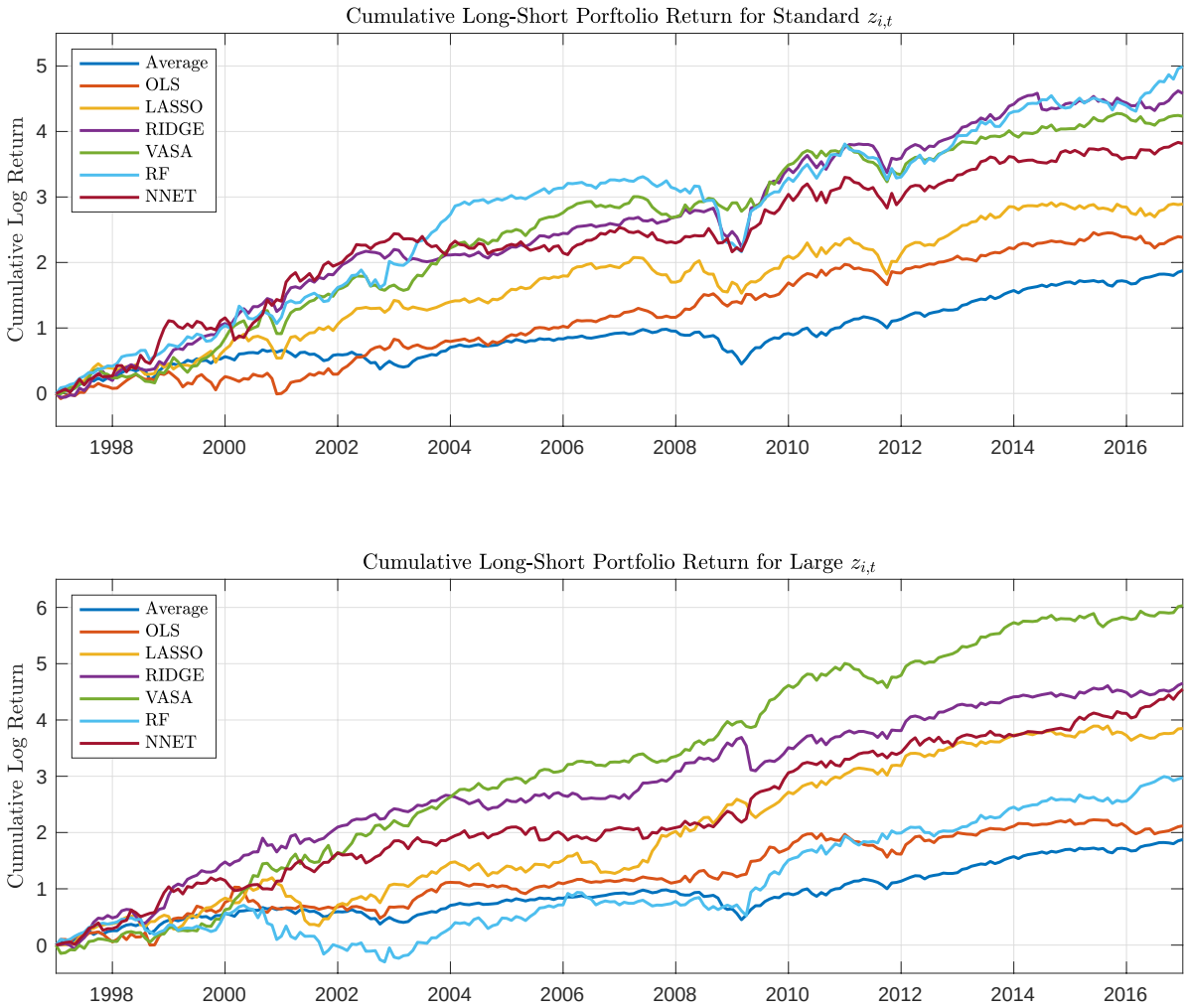

Figure 5: Cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts based on $\boldsymbol{z}_{i,t}^{\text{standard}}$ and $\boldsymbol{z}_{i,t}^{\text{large}}$. All portfolios are value-weighted.

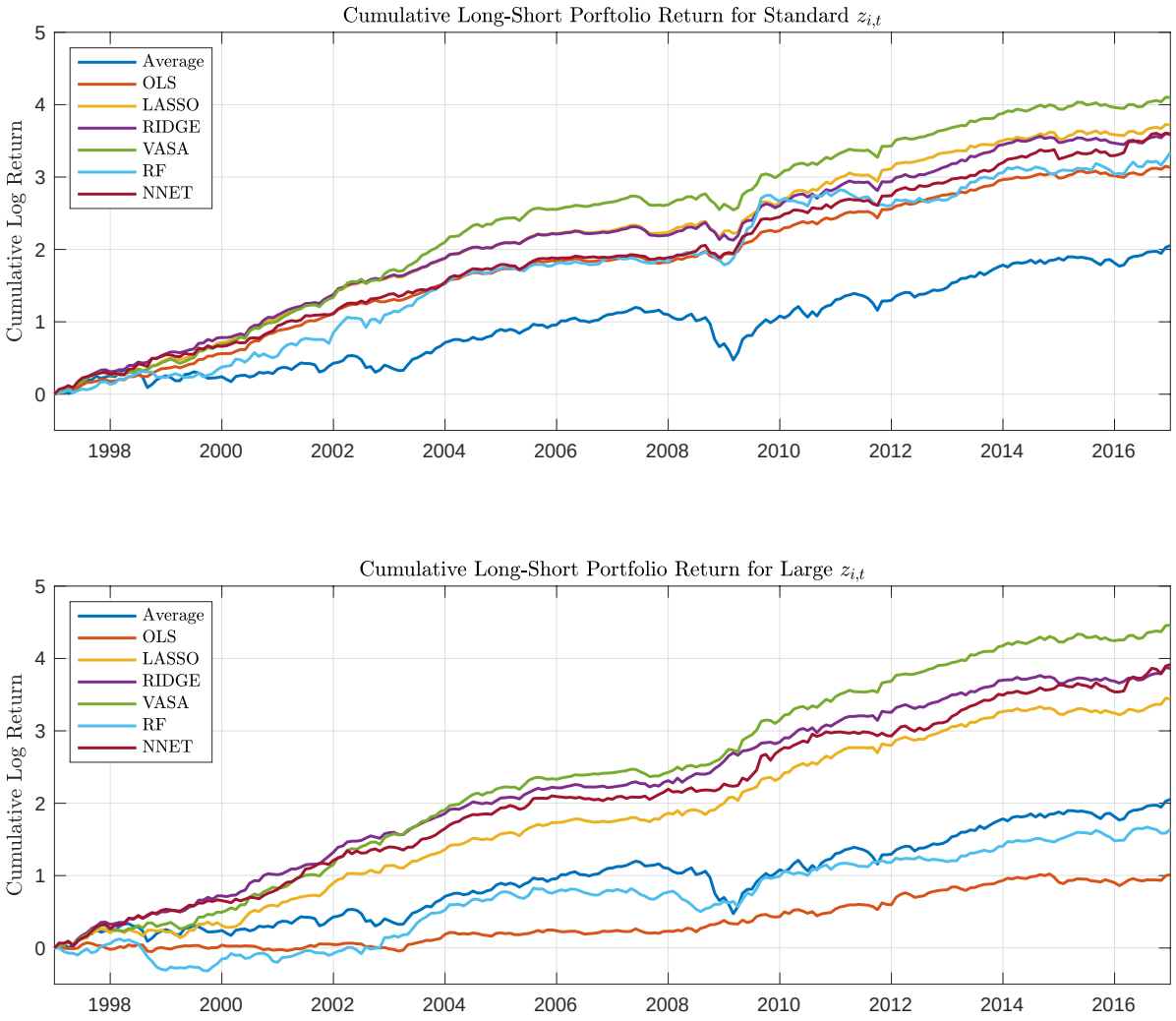

Figure 6: Cumulative log returns of portfolios sorted on out-of-sample machine learning return forecasts based on $\boldsymbol{z}_{i,t}^{\text{standard}}$ and $\boldsymbol{z}_{i,t}^{\text{large}}$ and efficient sorting.

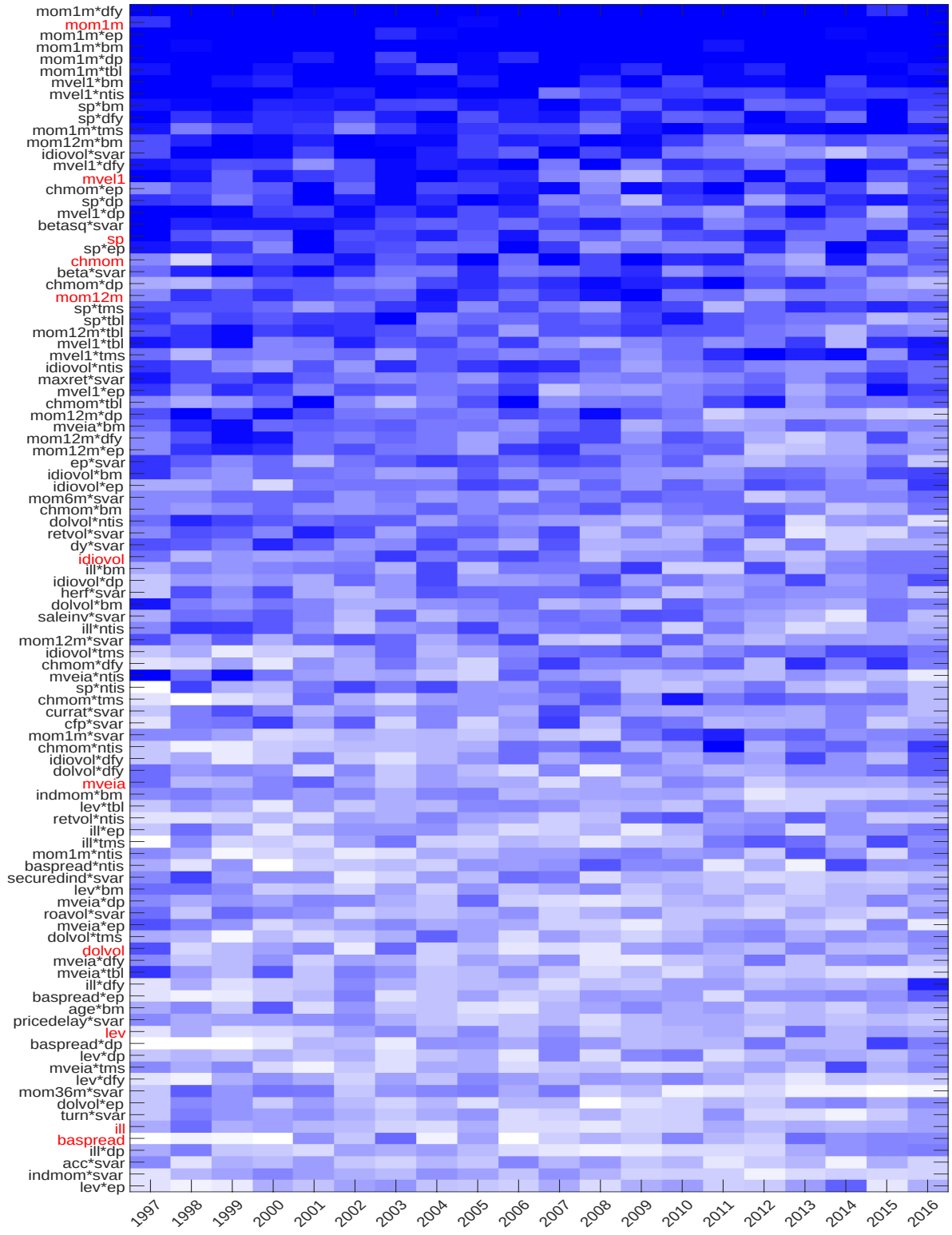

Figure 7: This figure describes how VASA ranks (over time) all the 904 covariates in terms of overall model importance. We plot only the 100 most important covariates and highlight in red the stock-level characteristics (without any macroeconomic interaction). Columns correspond to the year end of each of the 20 samples, and color gradients within each column indicate the most frequent (dark blue) to least frequent (white) covariates.

Base-case Setting: Comparison of Machine Learning Methods

	"Oracle"	Average	OLS	LASSO	RIDGE	VASA	RF	NNET
MED	8.80	3.07	8.39	**9.02**	9.00	8.74	8.40	6.88
AV	5.61	1.17	5.10	**5.81**	5.16	5.64	4.94	4.49
SD	28.27	29.37	28.47	**28.22**	28.44	28.28	28.47	28.71
P10	−30.96	−41.19	−33.71	−31.33	−33.62	**−31.06**	−34.85	−36.35
R^2_{OOS}	4.53	0.00	4.01	**4.74**	4.08	4.57	3.87	3.43

Table 1: This table presents summary statistics for the 100 out-of-sample $R^2_{OOS,i}$ in % for the base-case scenario. Additionally, the last row contains the global R^2_{OOS} in %. In the rows labeled MED, AV, P10, and R^2_{OOS} the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**.

Nonlinear Setting: Comparison of Machine Learning Methods

	"Oracle"	Average	OLS	LASSO	RIDGE	VASA	RF	NNET
MED	17.75	10.02	9.36	11.55	10.19	11.93	**16.59**	11.77
AV	12.60	4.80	6.12	6.62	6.26	6.76	**11.27**	6.69
SD	23.88	25.81	25.42	25.41	25.48	25.40	**24.11**	25.28
P10	−18.71	−28.81	−27.77	−26.85	−27.46	−25.73	**−19.24**	−23.83
R^2_{OOS}	11.66	3.75	5.08	5.63	5.25	5.77	**10.28**	5.72

Table 2: This table presents summary statistics for the 100 out-of-sample $R^2_{OOS,i}$ in % for the nonlinear scenario. Additionally, the last row contains the global R^2_{OOS} in %. In the rows labeled MED, AV, P10, and R^2_{OOS} the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**.

Sparsity: Comparison of Machine Learning Methods								
	"Oracle"	Average	OLS	LASSO	RIDGE	VASA	RF	NNET
	$K^* = 1$ Driving Covariate							
MED	6.30	2.12	3.87	6.12	3.79	**6.20**	5.01	4.46
AV	5.14	0.94	2.95	4.85	3.08	**4.86**	4.18	3.37
SD	**21.37**	22.27	21.90	21.48	21.80	21.52	21.53	22.04
P10	−23.15	−26.59	−26.67	**−23.16**	−25.05	−23.52	−23.71	−24.84
R^2_{OOS}	4.25	0.00	2.05	3.96	2.16	**3.97**	3.29	2.47
	$K^* = 3$ Driving Covariates							
MED	8.80	3.07	8.39	**9.02**	9.00	8.74	8.40	6.88
AV	5.61	1.17	5.10	**5.81**	5.16	5.64	4.94	4.49
SD	28.27	29.37	28.47	**28.22**	28.44	28.28	28.47	28.71
P10	−30.96	−41.19	−33.71	−31.33	−33.62	**−31.06**	−34.85	−36.35
R^2_{OOS}	4.53	0.00	4.01	**4.74**	4.08	4.57	3.87	3.43
	$K^* = 10$ Driving Covariates							
MED	14.99	6.05	14.10	**15.03**	14.02	14.79	13.30	13.12
AV	10.79	1.92	9.53	10.67	10.12	**10.68**	7.84	9.47
SD	21.40	24.93	21.70	21.66	21.69	**21.44**	22.50	21.78
P10	−20.73	−37.25	−23.45	**−20.10**	−21.35	−20.43	−24.90	−22.84
R^2_{OOS}	9.00	0.00	7.73	**8.89**	8.33	**8.89**	5.99	7.67

Table 3: This table presents summary statistics for the 100 out-of-sample $R^2_{OOS,i}$ in % for various levels of sparsity scenarios. Additionally, the last row contains the global R^2_{OOS} in %. In the rows labeled MED, AV, P10, and R^2_{OOS}, the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**.

Signal-to-Noise Ratio: Comparison of Machine Learning Methods								
	"Oracle"	Average	OLS	LASSO	RIDGE	VASA	RF	NNET
	$\theta_0 = 0.02$							
MED	8.80	3.07	8.39	**9.02**	9.00	8.74	8.40	6.88
AV	5.61	1.17	5.10	**5.81**	5.16	5.64	4.94	4.49
SD	28.27	29.37	28.47	**28.22**	28.44	28.28	28.47	28.71
P10	−30.96	−41.19	−33.71	−31.33	−33.62	**−31.06**	−34.85	−36.35
R^2_{OOS}	4.53	0.00	4.01	**4.74**	4.08	4.57	3.87	3.43
	$\theta_0 = 0.05$							
MED	35.14	9.53	34.61	**35.34**	34.62	35.05	34.61	34.19
AV	31.11	4.21	30.75	**31.26**	30.82	31.08	29.41	30.19
SD	25.77	41.14	25.90	**25.74**	25.85	25.80	26.83	26.25
P10	−2.67	−54.85	−2.41	**−2.32**	−2.59	−2.53	−6.24	−3.90
R^2_{OOS}	27.95	0.00	27.56	**28.11**	27.63	27.92	26.20	26.97
	$\theta_0 = 0.1$							
MED	61.68	−4.57	61.27	61.50	60.51	**61.63**	56.95	60.58
AV	59.95	−16.20	59.32	59.82	59.00	**59.98**	55.13	58.78
SD	14.10	52.23	14.41	14.16	14.63	**14.08**	16.39	14.72
P10	44.41	−88.70	44.29	**44.87**	43.60	44.69	35.93	40.97
R^2_{OOS}	65.03	0.00	64.47	64.91	64.21	**65.06**	60.92	64.02

Table 4: This table presents summary statistics for the 100 out-of-sample $R^2_{OOS,i}$ in % for various signal to noise ratios. Additionally, the last row contains the global R^2_{OOS} in %. In the rows labeled MED, AV, P10, and R^2_{OOS}, the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**.

No.	Acronym	Firm Characteristic	Literature
1	absacc	Absolute accruals	Bandyopadhyay et al. (2010)
2	acc	Working capital accruals	Sloan (1996)
3	aeavol	Abnormal earnings announcement volume	Lerman et al. (2008)
4	age	Years since first Compustat coverage	Jiang et al. (2005)
5	agr	Asset growth	Cooper et al. (2008)
6	baspread	Bid-ask spread	Amihud and Mendelson (1989)
7	beta	Beta	Fama and MacBeth (1973)
8	betasq	Beta squared	Fama and MacBeth (1973)
9	bm	Book-to-market	Rosenberg et al. (1985)
10	bmia	Industry-adjusted book-to-market	Asness et al. (2000)
11	cash	Cash holdings	Palazzo (2012)
12	cashdebt	Cash flow to debt	Ou and Penman (1989)
13	cashpr	Cash productivity	Chandrashekar and Rao (2009)
14	cfp	Cash flow to price ratio	Desai et al. (2004)
15	cfpia	Industry-adjusted cash flow to price ratio	Asness et al. (2000)
16	chatoia	Industry-adjusted change in asset turnover	Soliman (2008)
17	chcsho	Change in shares outstanding	Pontiff and Woodgate (2008)
18	chempia	Industry-adjusted change in employees	Asness et al. (2000)
19	chinv	Change in inventory	Thomas and Zhang (2002)
20	chmom	Change in 6-month momentum	Gettleman and Marks (2006)
21	chpmia	Industry-adjusted change in profit margin	Soliman (2008)
22	chtx	Change in tax expense	Thomas and Zhang (2011)
23	cinvest	Corporate investment	Titman et al. (2004)
24	convind	Convertible debt indicator	Valta (2016)
25	currat	Current ratio	Ou and Penman (1989)
26	depr	Depreciation / PP&E	Holthausen and Larcker (1992)
27	divi	Dividend initiation	Michaely et al. (1995)
28	divo	Dividend omission	Michaely et al. (1995)
29	dolvol	Dollar trading volume	Chordia et al. (2001)
30	dy	Dividend to price	Litzenberger and Ramaswamy (1982)
31	ear	Earnings announcement return	Kishore et al. (2008)
32	egr	Growth in common shareholder equity	Richardson et al. (2005)
33	ep	Earnings to price	Basu (1977)
34	gma	Gross profitability	Novy-Marx (2013)
35	grCAPX	Growth in capital expenditures	Anderson and Garcia-Feijóo (2006)
36	grltnoa	Growth in long term net operating assets	Fairfield et al. (2003)
37	herf	Industry sales concentration	Hou and Robinson (2006)
38	hire	Employee growth rate	Belo et al. (2014)
39	idiovol	Idiosyncratic return volatility	Ali et al. (2003)
40	ill	Illiquidity	Amihud (2002)
41	indmom	Industry momentum	Moskowitz and Grinblatt (1999)
42	invest	Capital expenditures and inventory	Moskowitz and Grinblatt (2010)
43	lev	Leverage	Bhandari (1988)
44	lgr	Growth in long-term debt	Richardson et al. (2005)
45	maxret	Maximum daily return	Bali et al. (2011)
46	mom12m	12-month momentum	Jegadeesh and Titman (1993)
47	mom1m	1-month momentum	Jegadeesh and Titman (1993)
48	mom36m	36-month momentum	Jegadeesh and Titman (1993)
49	mom6m	6-month momentum	Jegadeesh and Titman (1993)
50	ms	Financial statement score	Mohanram (2005)

Table 5: This table lists the 94 characteristics we use in the empirical study. We obtain the characteristics used by Gu et al. (2020) from Dacheng Xiu's webpage; see http://dachxiu.chicagobooth.edu. Note that data are collected in Green et al. (2017).

No.	Acronym	Firm Characteristic	Literature
51	mvel1	Size	Banz (1981)
52	mveia	Industry-adjusted size	Asness et al. (2000)
53	nincr	Number of earnings increases	Barth et al. (1999)
54	operprof	Operating profitability	Fama and French (2015)
55	orgcap	Organizational capital	Eisfeldt and Papanikolaou (2013)
56	pchcapxia	Industry adjusted change in capital exp.	Abarbanell and Bushee (1998)
57	pchcurrat	Change in current ratio	Ou and Penman (1989)
58	pchdepr	Change in depreciation	Holthausen and Larcker (1992)
59	pchgmpchsale	Change in gross margin - change in sales	Abarbanell and Bushee (1998)
60	pchquick	Change in quick ratio	Ou and Penman (1989)
61	pchsalepchinvt	Change in sales - change in inventory	Abarbanell and Bushee (1998)
62	pchsalepchrect	Change in sales - change in A/R	Abarbanell and Bushee (1998)
63	pchsalepchxsga	Change in sales - change in SG&A	Abarbanell and Bushee (1998)
64	ppchsaleinv	Change sales-to-inventory	Ou and Penman (1989)
65	pctacc	Percent accruals	Hafzalla et al. (2011)
66	pricedelay	Price delay	Hou and Moskowitz (2005)
67	ps	Financial statements score	Piotroski (2000)
68	quick	Quick ratio	Ou and Penman (1989)
69	rd	R&D increase	Eberhart et al. (2004)
70	rdmve	R&D to market capitalization	Guo et al. (2006)
71	rdsale	R&D to sales	Guo et al. (2006)
72	realestate	Real estate holdings	Tuzel (2010)
73	retvol	Return volatility	Ang et al. (2006)
74	roaq	Return on assets	Balakrishnan et al. (2010)
75	roavol	Earnings volatility	Francis et al. (2004)
76	roeq	Return on equity	Hou et al. (2015)
77	roic	Return on invested capital	Brown and Rowe (2007)
78	rsup	Revenue surprise	Kama (2009)
79	salecash	Sales to cash	Ou and Penman (1989)
80	saleinv	Sales to inventory	Ou and Penman (1989)
81	salerec	Sales to receivables	Ou and Penman (1989)
82	secured	Secured debt	Valta (2016)
83	securedind	Secured debt indicator	Valta (2016)
84	sgr	Sales growth	Lakonishok et al. (1994)
85	sin	Sin stocks	Hong and Kacperczyk (2009)
86	sp	Sales to price	Barbee et al. (1996)
87	stddolvol	Volatility of liquidity (dollar trading volume)	Chordia et al. (2001)
88	stdturn	Volatility of liquidity (share turnover)	Chordia et al. (2001)
89	stdacc	Accrual volatility	Bandyopadhyay et al. (2010)
90	stdcf	Cash flow volatility	Huang (2009)
91	tang	Debt capacity/firm tangibility	Almeida and Campello (2007)
92	tb	Tax income to book income	Lev and Nissim (2004)
93	turn	Share turnover	Datar et al. (1998)
94	zerotrade	Zero trading days	Liu (2006)

Table 6: Table 5 continued.

Monthly Out-Of-Sample Stock-level Prediction Performance (in %)								
	Average	OLS	OLS-3	LASSO	RIDGE	VASA	RF	NNET
				$\boldsymbol{z}_{i,t}^{\text{standard}}$				
MED	1.02	1.03	0.88	1.22	1.24	**1.26**	0.89	1.09
AV	1.12	1.03	0.89	1.28	1.29	**1.33**	0.95	1.16
SD	1.01	1.49	1.24	1.22	1.07	1.22	**0.95**	1.20
P10	−0.04	−0.53	−0.17	−0.09	**−0.00**	−0.04	−0.14	−0.23
R^2_{OOS}	0.81	0.87	0.77	0.95	0.95	**0.97**	0.77	0.82
				$\boldsymbol{z}_{i,t}^{\text{large}}$				
MED	1.02	−39.64	0.88	0.84	0.91	1.23	2.55	**2.59**
AV	1.12	−513	0.89	0.46	0.70	1.07	2.00	**2.66**
SD	**1.01**	2080	1.24	2.39	1.93	1.46	5.24	2.59
P10	−0.04	−607	−0.17	−2.54	−1.97	−0.56	−4.53	**0.36**
R^2_{OOS}	0.81	−194	0.77	0.93	1.07	1.10	**2.53**	2.30

Table 7: This table presents summary statistics for the 501 out-of-sample $R^2_{OOS,i}$ and the global R^2_{OOS} for the standard and large set of stock-level characteristics. In the rows labeled MED, AV, R^2_{OOS} and P10 the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**. Note that OLS-3 pre-selects size, book-to-market, and momentum as the only covariates.

Long-Short Portfolio Analysis (Equal-Weighted)								
	Average	OLS	OLS-3	LASSO	RIDGE	VASA	RF	NNET
	$\boldsymbol{z}_{i,t}^{\text{standard}}$							
Value	5.129	41.271	14.816	75.957	56.204	**91.181**	29.535	52.434
AV	9.504	20.025	16.354	23.892	23.273	**24.790**	19.975	22.843
SD	**16.104**	16.158	24.037	20.320	24.492	20.260	24.339	24.368
SR	0.590	**1.239**	0.680	1.176	0.950	1.224	0.821	0.937
Skew	−0.330	0.145	**0.554**	−0.234	0.087	−0.024	0.250	0.329
Kurt	2.433	1.975	2.973	0.745	3.338	**0.070**	1.500	0.984
	$\boldsymbol{z}_{i,t}^{\text{large}}$							
Value	5.129	10.377	14.816	47.378	111.362	**522.675**	34.154	49.571
AV	9.504	13.346	16.354	21.861	26.191	**34.271**	20.580	21.424
SD	**16.104**	18.097	24.037	21.966	21.903	23.156	24.049	18.848
SR	0.590	0.737	0.680	0.995	1.196	**1.480**	0.856	1.137
Skew	−0.330	**0.567**	0.554	−0.340	−0.353	0.416	0.492	0.002
Kurt	2.433	2.410	2.973	1.862	1.945	**0.481**	2.079	1.298

Table 8: Annualized performance measures for all models of the equal-weighted long-short portfolio. Value stands for the final portfolio value. AV denotes the average mean excess return and SD stands for standard deviation. By SR, we denote the Sharpe ratio. By Skew and Kurt, we denote skewness and excess kurtosis. All measures are annualized and based on 240 monthly out-of-sample returns from January 1997 until December 2016. In the rows labeled Value, AV, SR, and Skew, the largest number appears in **bold face**. In the rows labeled SD and Kurt, the lowest number appears in **bold face**.

Portfolio Analysis (Value-Weighted)								
	Average	OLS	OLS-3	LASSO	RIDGE	VASA	RF	NNET
	$z_{i,t}^{\text{standard}}$							
Value	6.520	10.841	54.507	18.030	97.577	68.906	**146.833**	45.408
AV	10.350	13.519	23.852	17.052	26.283	24.149	**29.597**	23.151
SD	**13.692**	17.678	27.646	22.503	25.947	24.052	30.176	28.553
SR	0.756	0.766	0.863	0.758	**1.013**	1.004	0.981	0.811
Skew	−0.448	0.075	0.422	−0.028	**1.302**	0.380	0.552	0.494
Kurt	1.095	1.892	1.602	**0.512**	10.335	1.221	4.153	1.123
	$z_{i,t}^{\text{large}}$							
Value	6.520	8.320	54.507	46.927	104.961	**416.884**	19.612	94.298
AV	10.35	12.851	23.852	22.462	26.414	**33.577**	18.761	26.030
SD	**13.692**	21.382	27.646	24.768	24.146	25.081	28.155	25.379
SR	0.756	0.601	0.863	0.907	1.094	**1.339**	0.666	1.026
Skew	−0.448	0.654	0.422	−0.249	−0.226	0.276	**0.903**	0.659
Kurt	1.095	2.314	1.602	1.312	3.784	**0.652**	6.139	3.355

Table 9: Annualized performance measures (in percent) for all models of the value-weighted long-short portfolio. Value stands for the final portfolio value; AV stands for average; SD stands for standard deviation; and SR stands for Sharpe ratio. All measures are based on 240 monthly out-of-sample returns from January 1997 until December 2016. In the rows labeled Value, AV, and SR the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**. Except for the Sharpe ratio, all numbers are expressed in percentage numbers.

Portfolio Analysis (Efficient Sorting)								
	Average	OLS	OLS-3	LASSO	RIDGE	VASA	RF	NNET
	$\boldsymbol{z}_{i,t}^{\text{standard}}$							
Value	7.802	22.889	2.009	41.230	36.100	**60.319**	28.174	36.352
AV	11.608	16.122	4.310	19.386	18.719	**21.426**	18.08	18.796
SD	16.055	**8.660**	12.778	11.547	11.547	12.495	16.395	11.967
SR	0.723	**1.862**	0.337	1.679	1.617	1.715	1.103	1.571
Skew	−0.365	0.461	−0.253	0.746	0.427	0.785	**1.342**	0.600
Kurt	2.458	2.546	**1.307**	3.879	3.294	3.282	4.004	3.116
	$\boldsymbol{z}_{i,t}^{\text{large}}$							
Value	7.802	2.759	2.009	31.133	47.593	**86.393**	5.161	50.233
AV	11.608	5.460	4.310	17.972	20.028	**23.215**	9.110	20.552
SD	16.055	**8.719**	12.778	11.617	10.69	12.166	13.355	12.974
SR	0.723	0.626	0.337	1.547	1.874	**1.908**	0.682	1.584
Skew	−0.365	0.525	−0.254	0.427	0.178	0.768	0.507	**0.941**
Kurt	2.458	2.084	1.307	**0.482**	0.503	1.681	0.839	3.436

Table 10: Annualized performance measures (in percent) for all models of the efficient sorting long-short portfolio. Value stands for the final portfolio value; AV stands for average; SD stands for standard deviation; and SR stands for Sharpe ratio. All measures are based on 240 monthly out-of-sample returns from January 1997 until December 2016. In the rows labeled Value, AV, and SR the largest number appears in **bold face**. In the row labeled SD the lowest number appears in **bold face**. Except for the Sharpe ratio, all numbers are expressed in percentage numbers.

A Distribution of the Subsampling Vector and Selection Matrix

We define the P-dimensional randomly generated subsampling vector V_b as $\{0,1\}^P$ such that $V_b'\mathbb{1} = K_b$. Hence, for $V_b = \{v_{b,1}, \ldots, v_{b,P}\}'$, $v_{b,l} = 1$ indicates that variable $l \in \{1, \ldots, P\}$ is selected, whereas $v_{b,l} = 0$ indicates that variable l is not selected. Therefore, to get V_b we draw from a set of $\binom{P}{K_b}$ vectors, each of size P containing exactly K_b ones and $P - K_b$ zeros. Hence, the probability of drawing a certain combination of the K_b ones is $\mathbb{P}[V_b = \mathrm{v}] = \frac{1}{\binom{P}{K_b}}$. This can be written as a special case of a multivariate hypergemetric distribution in the following way

$$V_b \sim \mathrm{HGeom}(B, K_b) \ ,$$

with

$$\mathbb{P}_{B,K_b}[V_{b,1} = \mathrm{v}_1, \ldots, V_{b,P} = \mathrm{v}_P] = \frac{\binom{B_1}{\mathrm{v}_1} \times \binom{B_2}{\mathrm{v}_2} \times \ldots \times \binom{B_P}{\mathrm{v}_P}}{\binom{P}{K_b}} \ ,$$

where

$$B_1 + \ldots + B_P = P, \quad \mathrm{v}_1 + \ldots + \mathrm{v}_P = K_b \quad \text{and} \quad K_b \leqslant P \ .$$

In our case we have that

$$B = \{1, \ldots, 1\} \quad \text{and} \quad \mathrm{v}_j \in \{0,1\} \ ,$$

Hence, we write in short

$$V_b \sim \mathrm{HGeom}(P, K_b), \quad \text{with } \mathbb{P}[V_b = \mathrm{v}] = \frac{1}{\binom{P}{K_b}} \quad \blacksquare \tag{A.1}$$

The variable selection matrix $\Lambda(V_b) \in \{0,1\}^{K_b \times P}$ is based on the P-dimensional subsampling vector as it transforms V_b in a $K_b \times P$ matrix which selects only the rows of $X \in \mathbb{R}^{P \times N}$ for the in V_b indicated variables; see Equation (2.12). For example take $P = 3$, $N = 4$ and (randomly) choose two out of the three variables ($K = 2$). Then, V_b can be $\{1,1,0\}'$, $\{0,1,1\}'$ or $\{1,0,1\}'$ all with equal probability of $\frac{1}{3}$. Now focus on the last realization where the first and the third

variable is selected and the second is dropped $V_b = \{1, 0, 1\}'$:

$$
\begin{aligned}
\tilde{X}_b &= \Lambda(V_b)X \ , \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{bmatrix} \ , \\
&= \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{bmatrix} \ .
\end{aligned}
$$